الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Dr. Tahar Moulay SAIDA
Faculté : Technologie
Département : Informatique

جامعة د. الطاهر مولاي سعيدة
كلية: التكنولوجيا
قسم: الإعلام الآلي

Mémoire de fin d'étude de Master en Informatique
Option : Modélisation Informatique des Connaissances et du Raisonnement

Thème

Modélisation du processus de recherche d'information par les éléphants d'Asie sociaux

Présenté par :
Mr Djamel MOSTEFAÏ
Mr Omar FEKIR

Encadré par :
Mr Mohamed Réda HAMOU

Promotion : Juin 2018

Résumé

Le but de ce mémoire est de consolider l’expression suivante : les techniques bio-inspirées représentent une solution fiable pour bâtir des systèmes performants pour les tâches de recherche d’information.

Pour prouver cette allégation, nous avons modéliser le processus de recherche d'information par les éléphants d'Asie sociaux. Cette Approche permet de résoudre différent problèmes d’application de la recherche d’information tel que la catégorisation de textes.

Après l’évaluation de notre approche, les résultats obtenu montre qu'on peut s'inspirer de la nature pour bâtir des systèmes performants.

Mots-clés : Bio-inspirée ; Métaheuristique ; Recherche d’Information ; la catégorisation de textes ;

Abstract

The purpose of this dissertation is to consolidate the following expression: bio inspired techniques represent a reliable solution for building efficient systems for information retrieval tasks. To prove this claim, we modeled the process of information retrieval by Asian social elephants. This Approach solves various problems of application of information retrieval such as categorization of texts.

After evaluating our approach, the results obtained show that we can take inspiration from nature to build efficient systems.

Keywords: Bio-inspired; Meta-heuristic; Information retrieval; Texts categorization;

ملخص

الغرض من هذه المذكرة هو دمج التعبير التالي: التقنيات المستوحاة من الحيوية تمثل حلا يعتمد عليه لبناء أنظمة فعالة لمهام البحث عن المعلومات. لإثبات هذا الادعاء، قمنا بتصميم عملية البحث عن المعلومات بواسطة الأفيال الاجتماعية الآسيوية. هذا النهج يحل المشاكل المختلفة لتطبيق البحث عن المعلومات مثل تصنيف النصوص.

بعد تقييم نهجنا، تظهر النتائج التي تم الحصول عليها أننا يمكن أن نستلهم من الطبيعة لبناء أنظمة فعالة.

الكلمات المفتاحية: الإيحاء من الطبيعة، الأدلة العليا، تصنيف النصوص.

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur Mr Mohamed Réda HAMOU, son précieux conseil et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nous remercions, aussi, les enseignants du département de l'informatique qui nous ont soutenus tout au long de notre cursus.

A nos familles et nos amis qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Merci

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Le but de la Recherche d'Information est de faciliter l'accès à l'information et le rendre une tâche simple. Hors que derrière cette tache se cache tout un processus compliqué qui comporte trois grandes axes : l'indexation des documents, la correspondance (pertinence) et enfin l'interrogation qui englobe la représentation, l'analyse et la reformulation des requêtes. Les systèmes de recherche d'informations sont utilisés dans différents domaines et en général dans le domaine de la classification.

L'évaluation de ces systèmes se repose sur plusieurs critères : le temps de réponse, la pertinence des résultats, la qualité de la présentation, etc. Comment définir un système qui satisfait les besoins de l'utilisateur ? Cette satisfaction se traduit par le temps que prenne ce système entre la demande envoyée et les documents retournés.

Plusieurs techniques ont été utilisées pour que ces systèmes fournissent la pertinence attendue par l'utilisateur. Parmi les plus connues, on trouve celles basées sur des métaheuristiques, et principalement les algorithmes inspirés de la nature. Cette dernière qui nous donne plein d'exemples pour résoudre des problèmes complexes en informatique à travers des phénomènes divers groupés par le champ biologique dont il a inspiré chacun d'eux.

Dans le contexte de notre travail, on se concentre sur les algorithmes inspirés des éléphants d'Asie qui sont des animaux intelligents et qui vivent en groupes.

L'objectif est de produire un nouveau classificateur basé sur les résultats des études de Byrne & al dans [32] qui ont prouvés que les éléphants d'Asie, les chimpanzés et les dauphins ont une vie sociale dynamique et complexe basée sur l'esprit du groupe.

Ce document est organisé comme suit :

Dans le premier chapitre, on présente quelques définitions de la recherche d'information et les concepts liés à ce terme. Nous décrivons ainsi le processus de recherche d'information. Le deuxième chapitre port sur la bio-inspiration et les métaheuristiques et les approches les plus connues qui existent dans la littérature. Dans le troisième chapitre, on décrit le modèle basé sur les éléphants d'Asie. Et enfin, dans le chapitre 4, nous présentons notre expérimentation qui a pour but de concevoir un classificateur basé sur le modèle d'éléphant d'Asie pour la recherche d'information.

Chapitre premier
Recherche d’information

I. Recherche d'information

I.A. Introduction

Historiquement, la croissance du volume de données textuelles comme les livres et les articles dans les bibliothèques durant des siècles a imposé une définition des mécanismes efficaces pour les localiser. D'où la naissance de la Recherche d'Information qui est un ancien domaine datant des années 1940 [36]. Une des premières définitions a été introduite par Calvin Mooers en 1950 où il définit la Recherche d'Information comme une discipline informatique traitant le problème d'accès à l'information pertinente dans une masse de données importante [4]. La Recherche d'Information est l'ensemble d'opérations, de méthodes et de procédures qui permettent de retrouver des documents pertinent à partir d'une collection de documents pouvant répondre à une question sur un sujet précis [10].

L'opérationnalisation de la Recherche d'Information est réalisée par des outils informatiques appelés systèmes de recherche d'information (SRI).

Systèmes de Recherche d'Information, il s'agit d'un mécanisme de gestion qui sert d'intermédiaire entre un utilisateur et une collection d'informations dans le but de satisfaire le besoin en information de l'utilisateur [1].

I.B. Recherche d'information

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie qui ont toujours eu le souci de faire des représentations de documents afin de récupérer des informations à travers la construction d'index. L'informatique a permis le développement d'outils pour traiter l'information et établir la représentation des documents au moment de leur indexation, ainsi que pour rechercher des informations.

I.B.1. Définition

Plusieurs définitions de la recherche d'information ont émergé ces dernières années, nous citons dans ce contexte les trois définitions suivantes :

I.B.1.a. Définition 1

La recherche d'information est une activité dont la finalité est de localiser et de délivrer des granules documentaires à un utilisateur en fonction de son besoin en informations [27].

I.B.1.b. Définition 2

La recherche d'information est une branche de l'informatique qui s'intéresse à l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information [12].

I.B.1.c. Définition 3

La recherche d'information est une discipline de recherche qui intègre des modèles et des techniques dont le but est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information [33].

Toutes ces définitions partagent l'idée que la RI vise à extraire d'un document ou d'un ensemble de documents des informations pertinentes qui reflètent un besoin en information.

I.B.2. Concepts de base de la RI

La recherche d'information est considérée comme l'ensemble des techniques de sélection à partir d'une collection de documents, ceux qui sont susceptibles de répondre aux besoins de l'utilisateur. La gestion de ces informations implique le stockage, la recherche et l'exploration de documents pertinents. De ce contexte, plusieurs concepts clés peuvent être définis, nous avons donc trouvé utile de les clarifier.

I.B.2.a. Document

Le document est l'information de base d'une collection de documents. L'information élémentaire, appelée aussi granule de document, peut représenter tout ou une partie d'un document [36]. C'est l'élément essentiel dans un système de Recherche d'Information, un document est considéré comme un support physique de l'information, qui peut être du texte, une page Web ou du multimédia.

Dans le cas d'un document texte on peut le représenter selon deux vues [28] :

- La vue sémantique : ou la vue selon le contenu, elle se concentre sur l'information véhiculée dans le document ;
- La vue logique : elle définit la structure logique du document.

I.B.2.b. Collection de documents

La collection de documents (ou fond documentaire) constitue l'ensemble des informations utilisables et accessibles. Il se compose d'un ensemble de documents. Dans le cas général et par souci d'optimalité, la base de données constitue des représentations simplifiées mais suffisantes de ces documents. Ces représentations sont étudiées de telle sorte que la gestion (ajout d'un document supprimé) ou l'interrogation (recherche) de la base de données se fasse dans les meilleures conditions de coût.

I.B.2.c. Besoin en information

La notion de besoin en information à la recherche d'information est souvent assimilée au besoin de l'utilisateur. Trois types de besoins d'utilisateurs ont été définis par :

- **Besoin vérificatif** : L'utilisateur essaie de vérifier le texte avec les données connues qu'il possède déjà. Il cherche donc une information particulière, et sait même comment y accéder. La recherche d'un article sur Internet à partir d'une adresse connue serait un exemple d'un tel besoin. Un autre exemple serait de rechercher la date de publication d'un livre dont la référence est connue. Une exigence de type vérification est dite stable, c'est-à-dire qu'elle ne change pas pendant la recherche.
- **Besoin thématique connu** : l'utilisateur cherche à clarifier, revoir ou trouver de nouvelles informations dans un sujet et un domaine connus. Un besoin de ce type peut être stable ou variable : il est en effet tout à fait possible que le besoin de l'utilisateur soit affiné lors de la recherche. Le besoin peut également être exprimé de façon incomplète, c'est-à-dire que l'utilisateur n'énonce pas nécessairement tout ce qu'il sait dans sa requête mais seulement un sous-ensemble. C'est ce qu'on appelle dans la littérature l'étiquette (le label).
- **Besoin thématique inconnu** : Cette fois, l'utilisateur recherche de nouveaux concepts ou de nouvelles relations en dehors des sujets ou des domaines familiers. Le besoin est intrinsèquement variable et est toujours incomplètement exprimé.

I.B.2.d. Requête

La requête est l'expression des besoins en information de l'utilisateur selon le formalisme d'interrogation d'un système de recherche d'information [1]. Elle représente

l'interface entre le SRI et l'utilisateur. Différents types de langages d'interrogation sont proposés dans la littérature. Une requête est un ensemble de mots-clés, mais elle peut être en langage naturel, booléen ou graphique.

I.B.2.e. Modèle de représentation

Un modèle de représentation est un processus permettant d'extraire d'un document ou d'une requête, une description qui couvre au mieux son contenu sémantique. Ce processus de conversion s'appelle l'indexation. Le résultat de l'indexation est le descripteur du document ou de la requête, qui est une liste de termes ou groupes de termes (concepts), significatifs pour l'unité textuelle correspondante, avec laquelle les poids sont généralement associés, pour différencier leurs degrés les uns des autres. Représentativité du contenu sémantique de l'unité en question. L'ensemble des termes reconnus par le SRI est stocké dans une structure appelée dictionnaire constituant le langage d'indexation. Ce type de langage garantit le rappel des documents lorsque la requête utilise dans une large mesure les termes du dictionnaire. D'un autre côté, il existe un risque important de perte d'information lorsque la demande s'éloigne de ce vocabulaire.

I.B.2.f. Modèle de recherche

Il représente le modèle du noyau d'un SRI. Il comprend la fonction de décision fondamentale qui permet d'associer à une requête, tous les documents pertinents à restituer. Il est utilisé pour la recherche d'information lui-même et est étroitement lié au modèle de représentation des documents et des requêtes.

I.B.3. Pertinence

Est une notion cruciale qui mesure le degré de ressemblance entre la requête et les documents renvoyés en se référant aux deux concepts : bruit et silence. Tel que, le silence correspond aux documents pertinents qui n'apparaissent pas dans le résultat de la recherche, alors que le bruit correspond aux documents ramenés en réponse, mais qui ne sont pas pertinents par rapport à la question posée [6].

I.B.4. Modèles de recherche d'informations

Un modèle de RI a pour rôle de fournir une formalisation du processus de RI et un cadre théorique pour la modélisation de la mesure de pertinence. Il existe un grand nombre de modèles de RI textuels développés dans la littérature. Ces modèles ont, en commun, le vocabulaire d'indexation basé sur le formalisme du mot-clé et diffèrent principalement par le modèle de correspondance de requête-document. Le vocabulaire de l'index $V = t_i$, $i \in \{1, ..., n\}$ est constitué de n termes (mots) ou racines de mots qui apparaissent dans les documents.

Selon Baeza-Yates [31], un modèle de RI est défini par un quadruplet (D, Q, F, R (q, d)) : où

- D est l'ensemble des documents
- Q est l'ensemble des requêtes
- F est le schéma du modèle théorique de représentation des documents et des requêtes
- R (q, d) est la fonction de pertinence du document d à la requête q.

Nous présentons dans ce qui suit les principaux modèles de RI : le modèle booléen, le modèle vectoriel et le modèle probabiliste.

I.B.4.a. Modèle booléen

Le modèle booléen [15] est basé sur la théorie des ensembles. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots-clés. Chaque document est représenté par une conjonction logique de termes non pondérés constituant l'index du document. Un exemple de représentation d'un document est le suivant : $d = t_1 \wedge t_2 \wedge t_3 \ldots \wedge t_n$.

Une requête est une expression booléenne dont les termes sont connectés par des opérateurs logiques (OR, AND, NOT) pour effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme. Un exemple de représentation d'une requête est le suivant : $q = (t_1 \wedge t_2) \vee (t_3 \wedge t_4)$.

La fonction de correspondance est basée sur l'hypothèse de présence / absence des termes de la requête dans le document et vérifie si l'index de chaque document d_j implique l'expression logique de la requête q. Le résultat de cette fonction est donc binaire est décrit comme suit : RSV (q, d) = {1,0}.

I.B.4.b. Modèle vectoriel

Dans ces modèles [15], la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel. Le modèle vectoriel représente les documents et les requêtes par vecteurs d'un espace à n dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation.

L'indice d'un document d_j est le vecteur = ($w_{1,j}$, $w_{2,j}$, $w_{3,j}$, .., $w_{n,j}$), où $w_{k,j} \in [0\ 1]$ désigne le poids du terme t_k dans le document d_j. Une requête est également représentée par un vecteur = ($w_{1,q}$, $w_{2,q}$, $w_{3,q}$, .., $w_{n,q}$), où $w_{k,q}$ est le poids du terme t_k dans la requête q. La fonction de mappage mesure la similarité entre le vecteur de requête et les vecteurs de document. Une mesure classique utilisée dans le modèle vectoriel est le cosinus de l'angle formé par les deux vecteurs : RSV (q, d) = cos q d.

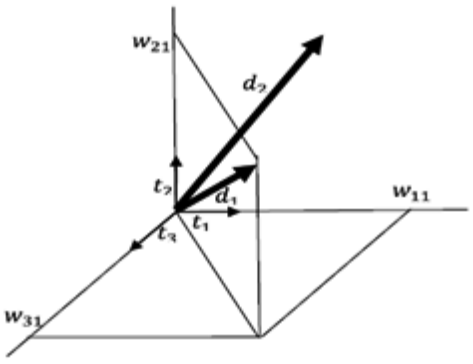

Figure I.1 – La représentation des documents dans l'espace d'indexation vectoriel

Plus les vecteurs sont similaires, plus l'angle formé est petit et plus le cosinus de cet angle est grand. Contrairement au modèle booléen, la fonction de correspondance évalue une correspondance partielle entre un document et une requête, ce qui vous permet de trouver des documents qui ne reflètent pas la requête comme approximative. Les résultats peuvent être classés par ordre décroissant de pertinence.

I.B.4.c. Modèle probabiliste

Ce modèle est basé sur le calcul de la probabilité de pertinence d'un document pour une requête [34][14][23]. Le principe de base est de récupérer des documents qui ont une forte probabilité d'être pertinents en même temps, et une faible probabilité d'être hors de propos.

Compte tenu d'une requête utilisateur Q et d'un document D, il s'agit de calculer la probabilité de pertinence du document pour cette requête.

Il y a deux possibilités : R, D est pertinent pour q et D, non pertinent pour q. Les documents et les requêtes sont représentés par des vecteurs booléens dans l'espace à n dimensions. Un exemple de représentation d'un document d_j et d'une requête q est le suivant :

$d_j = (w_{1,j}, w_{2,j}, w_{3,j}, .., w_{n,j})$,

$q = (w_{1,q}, w_{2,q}, w_{3,q}, .., w_{n,q})$. Avec $w_{k,j} \in [0\ 1]$ et $w_{k,q} \in [0\ 1]$.

La valeur de $w_{k,j}$ (resp., $W_{k,q}$) indique si le terme t_k apparaît ou non dans le document d_j (resp. q)

Le modèle probabiliste évalue la pertinence du document d_j pour la requête q. Un document est sélectionné si la probabilité que le document d soit pertinent, notée p(R/D), est supérieure à la probabilité que d est non pertinent pour q, noté p(R/D) où R est l'événement de pertinence et est l'événement de non-pertinence.

Le score d'appariement entre le document d et la requête Q, noté RSV (Q, D) est donné par : $(,) = \frac{(/)}{(/)}$

Ces probabilités sont estimées par des probabilités conditionnelles selon qu'un terme de requête est présent, dans un document pertinent ou dans un document non pertinent. Cette mesure de similarité entre la demande et les documents peut être calculée par différentes formules. Ce modèle a donné lieu à de nombreuses extensions. Il est à l'origine du système OKAPI. Le modèle Okapi BM25 a été développé par Robertson en 1994 dans lequel le calcul du poids d'un terme dans un document intègre des aspects relatifs à la fréquence locale des termes, à leur rareté et à la longueur des documents :

$$W(t,d) = \frac{f_{t,d} * (k_1 + 1)}{k_1 * \left((1-b) + b * \frac{dl}{avdl}\right) + f_{t,d}} * \log\left(\frac{N - df(t,C) + 0.5}{df(t,C) + 0.5}\right)$$

Avec $f_{t,d}$ est la fréquence du mot t dans le document d, dl est la taille du document (le nombre total d'occurrences de mots). $df(t,C)$ est le nombre de documents de la collection C contenant t. $k_1$et b sont des constantes qui dépendent des collections des tests ainsi que du type des requêtes. $avdl$ représente le moyen de longueur de tous les documents dans la collection C.

I.C. Les systèmes de recherche d'informations

I.C.1. Définition

Un système de recherche d'informations (*SRI*) est un système informatique qui vous permet de retourner à partir d'un ensemble de documents, ceux dont le contenu correspond le mieux aux besoins en information d'un utilisateur, exprimé à l'aide d'une requête.

Un SRI comprend un ensemble de procédures et d'opérations qui permettent la gestion, le stockage, l'interrogation, la recherche, la sélection et la représentation de cette masse d'informations.

I.C.2. Le processus de recherche d'information

Le but de la Recherche d'Information est de donner l'illusion que l'accès à l'information est une tâche simple qui se réalise en quelques clics, sauf que derrière cette simplicité se cache tout un processus compliqué, minutieux et robuste. Ci-dessous le modèle basique de ce processus. En fait le processus de la Recherche d'Information comporte trois grandes fonctions:

1. Fonction de l'indexation des documents : on transpose le corpus manipulé par le système de Recherche d'Information en corpus indexé ;
2. Fonction de correspondance (similarité) ;
3. Fonction d'interrogation : représentation, analyse et reformulation des requêtes.

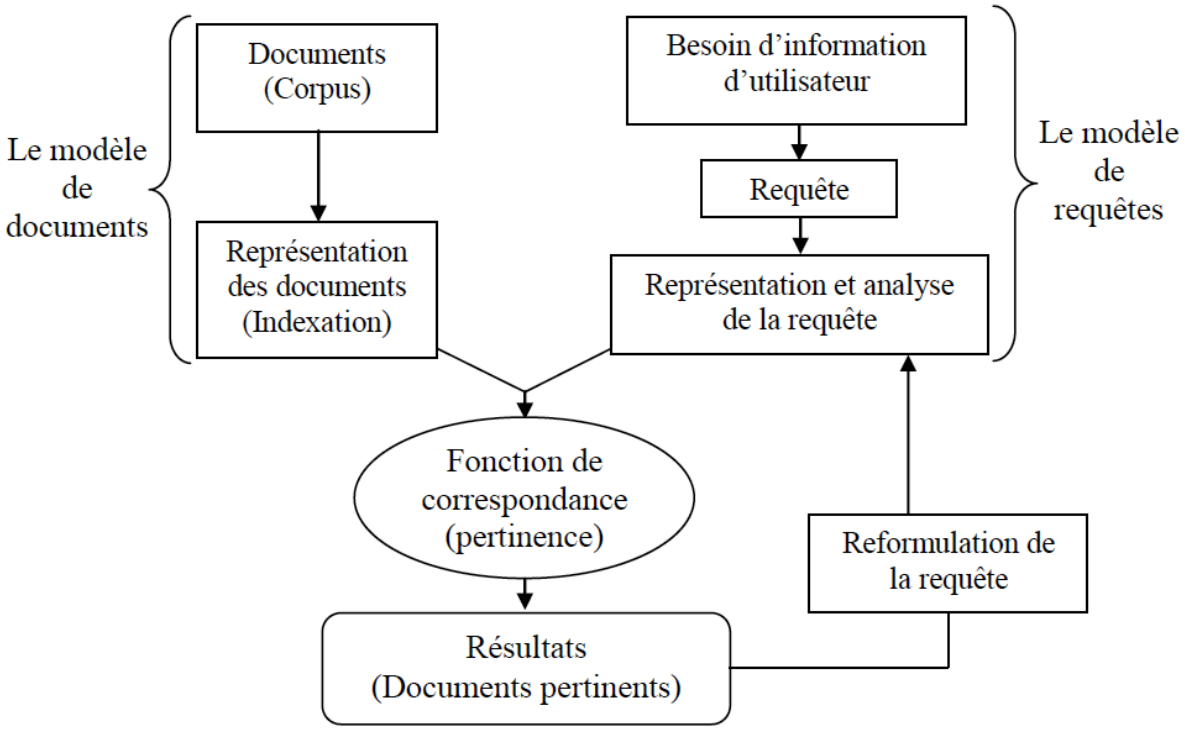

Figure I.2 – Modèle basique de processus de la Recherche d'Information

Le système de Recherche d'Information doit adopter un modèle de représentation, afin de synchroniser le modèle de document avec le modèle de requête (Voir *Figure I.2*), rappelons que le modèle de document est le modèle de transposition d'un document en un document, de même le modèle de requête est le modèle de la transformation du besoin utilisateur en requête.

I.C.3. Evaluation d'un système de recherche d'information

L'évaluation des systèmes de recherche d'information est un sujet de recherche important dans les sciences de l'information. Elle peut concerner plusieurs critères : le temps de réponse, la pertinence des résultats, la qualité de la présentation, etc. Le critère le plus important est sans aucun doute celui qui mesure la capacité du système à satisfaire le besoin d'information de l'utilisateur, cette satisfaction traduit dans une correspondance forte entre la demande envoyée et les documents retournés. Dans ce contexte, des campagnes d'évaluation ont été mises en place depuis les années 1960 telles que TREC (Text REtrieval Conference), CLEF (Cross-Language Evaluation Forum), INEX et AMARYLLIS, pour juger de l'efficacité

de ces systèmes et ainsi améliorer leurs performances en termes de technologie mais aussi en fonction des attentes des utilisateurs.

Les campagnes d'évaluation permettent d'évaluer plusieurs systèmes de Recherche d'Information par des collections différentes, afin de valider les différents modèles mis en œuvre, et de comparer les systèmes. Les objectifs essentiels des campagnes sont les suivants :

- Encourager la Recherche d'Information sur de grandes collections fermées ;
- Cordonner entre l'industrie, l'académie et le gouvernement par la mise en place d'une aire ouverte d'échange d'idées sur la recherche ;
- Rendre accessibles les nouvelles techniques d'évaluation pour les industriels et les académiciens.

I.D. Exemples en recherche d'information

La recherche d'information occupe une place importante aujourd'hui. Elle se pratique sur les documents de toute nature détenant une structure identifiée à travers des éléments et elle est basée sur des modèles et des algorithmes pour extraire le document recherché.

Les utilisateurs et les professionnels forment les deux grands groupes de personnes liées à la recherche d'information. Pour les utilisateurs, la recherche d'information devient une activité journalière. Pour la plupart, ce sont des recherches sur le web qui se font de manière intuitive et à l'aide d'appareils électroniques. L'utilisation généralisée des moteurs qui incitent les professionnels à trouver des améliorations et à faire évoluer la recherche d'information. Plusieurs tâches se font sur les résultats obtenus après avoir effectué la recherche, sur les données organisées et recherchées par le système :

Trouver : des documents pertinents sur des requêtes usagers qui peuvent être très vagues

Filtrer : détecter des points d'intérêt

Classifier : catégoriser en utilisant des éléments

Questions : retourner une réponse spécifique

Les moteurs de recherche représentent la mise en pratique de la recherche d'information. Pour récupérer les données, des méthodes variées peuvent être utilisées. En plus des problématiques apportées par la recherche d'information, il existe quelques fonctionnalités essentielles aux moteurs de recherche telles que la performance (temps de réponse), la rapidité, et la pratique.

On trouve, dans ce domaine, beaucoup d'exemples sur la recherche d'information, on cite les plus répandus :

I.D.1. Recherche d'information sur le web

Bill Clinton, ancien président américain a dit que « *Quand j'ai pris mes fonctions, seuls les physiciens nucléaires avaient déjà entendu parler de ce qu'on appelle le World Wide Web WWW), mais maintenant, même mon chat a sa propre page* ».

Dans la dernière décennie, le WWW est devenue une marchandise vitale utilisé sur une base quotidienne par un nombre croissant d'utilisateurs. Il représente un moyen extrêmement polyvalent et économique pour effectuer des tâches telles que la communication, le commerce et les affaires, les loisirs et le divertissement, la diffusion de la culture, et l'accès à

l'information. ...etc. La quantité des données stockées et partagées sur le Web et d'autres référentiels de documents ne cesse d'augmenter de façon persistante et abrupte. Malheureusement, cette croissance résulte des difficultés et des problèmes bien connus quand il vient à trouver les informations pertinentes. Mitchell Kapor[1] a dit dans son livre « Developing the national communications and information infrastructure» qu'obtenir des informations pertinentes à partir du web est comme prendre une boisson à partir d'une bouche d'incendie [26]. Il y a plusieurs arguments pourquoi la RI sur le web est si inefficace. Tout d'abord, le web est vaste, distribué, un espace accessible au grand public et contient essentiellement des données non structurées qui compliquent considérablement les tâches de RI.

De nombreuses applications qui gèrent l'information sur le web seraient totalement insuffisantes sans le soutien de la technologie de RI. Comment pourrions-nous trouver des informations sur le web s'il n'y avait pas de moteurs de recherche ? Comment pourrions-nous gérer nos e-mails sans le filtrage de spam ? Comment pourrions-nous protéger nos maisons et magasins à distance sans les systèmes de vidéosurveillance à travers le web ? À la lumière de ce qui précède, tous ces détails ont donné naissance à plusieurs problèmes que nous entreprenons dans cette thèse et qui sont colligés en quatre classes selon la tâche de recherche d'information dans le web.

I.D.1.a. Recherche ad hoc (RA)

La RA est le processus de rechercher les informations pertinentes à partir d'un amas de documents statique pour répondre à un besoin d'information posé par l'utilisateur sous forme d'une requête. Les problèmes liés à cette tâche sont nombreux :

I.D.1.a.i Interrogation

La formulation conventionnelle de la requête en utilisant simplement un ensemble de mots-clés n'aide pas l'utilisateur à articuler toutes les contraintes inévitables autour de ses besoins. Par exemple un utilisateur entre la requête « **les soins de la peau** », si cet utilisateur est un adolescent alors il est automatiquement intéressé par les documents de la manipulation d'un problème d'acné. Par contre, s'il est plus âgé alors il est intéressé par les documents de soin sur les rides. Ce qui nécessite que l'utilisateur précise son âge. Un autre exemple, celui d'un utilisateur qui cherche les documents qui parlent des différents sports sauf le football, dans ce cas une contrainte supplémentaire est incontestable pour aider l'utilisateur à définir cette exigence.

I.D.1.a.ii La qualité des résultats

Les systèmes de RA basés sur des modèles de recherche classique ne fournissent pas des bons résultats parce qu'ils retournent beaucoup de faux positifs[2] et beaucoup de faux négatifs[3]. Concernant les modèles booléens et vectoriels, cela est dû à : l'indépendance mutuelle dans le calcul des poids des termes, le mauvais choix de la mesure de similarité, l'imprécision dans la représentation des documents et l'incertitude dans le classement des résultats. Toutefois, concernant le modèle probabiliste cela est dû à la nécessité de deviner la séparation initiale des documents dans des ensembles pertinents et non-pertinents.

[1] Mitchell Kapor est un pionnier de l'industrie de l'informatique personnelle.

[2] Documents réellement non pertinents et le système les a classées comme pertinents.

[3] Documents réellement pertinents et le système les a classés comme non pertinents.

I.D.1.a.iii Recherche croisée

Les systèmes de RA classique sélectionnent uniquement les documents écrits en langue similaire à la requête. Par exemple nous tapons la requête en français « aimer » dans Google qui signifie « love» en anglais. Dans ce cas, Google cherche seulement les documents français contenant le mot « aimer », et ignore les documents écrits avec les autres langues. Cette situation exacerbe la nécessité de trouver un moyen de récupérer des informations à travers les frontières linguistiques.

I.D.1.a.iv Retour de pertinence

Les utilisateurs peuvent récupérer quelques documents pertinents en réponse à leurs requêtes, mais jamais tous les documents pertinents. Par contre, une fois que le système présente à l'utilisateur un ensemble initial de solutions, l'utilisateur peut indiquer les documents qui contiennent des informations utiles. Pour cela il est évident de prendre le jugement humain en considération afin d'améliorer la satisfaction des utilisateurs.

I.D.1.a.v La visualisation des informations

La majorité des systèmes de RA présentent les documents pertinents sous forme d'une liste, ce qui rend la visualisation de tous les résultats difficiles. Par exemple le moteur de recherche Google renvoie les documents pertinents pour une requête comme une liste de plusieurs pages, où la plupart des utilisateurs ne voient que les 2 ou 3 premières pages et ignorent les autres.

I.D.1.a.vi Recherche ad hoc privée

Ces dernières années le web et la vie privée sont deux mondes incompatibles parce que le respect de la vie privée est mis à mal par les moteurs de recherche, les réseaux sociaux, et les sites publicitaires. Par exemple un jour je cherchais des maisons de vacances sur Google et après lorsque je me suis connecté sur Facebook j'ai reçu des annonces sur les locations de vacances dans la ville que je cherchais. Un autre exemple plus clair, lorsque nous envisageons un service de stockage Cloud dans un hôpital utilisé par un système de RA pour répondre aux requêtes des patients sur des symptômes spécifiques ce qui exige aux utilisateurs de délivrer toutes ou partie des données personnelles. Malheureusement, ce service peut garder des traces sur les requêtes et les activités de chaque patient. C'est pour cette raison que la vie privée est devenue un problème d'intérêt majeur pour les systèmes de RA qui n'a pas encore été correctement pris en compte.

I.D.1.b. Filtrage d'information

Le filtrage[4] d'information consiste à identifier les documents pertinents et rejeter les documents non pertinents dans les flux d'informations entrants (le flux peut être message, image, vidéo, texte, ou page web. . . .etc.) pour répondre au profil de l'utilisateur. Ce principe peut être appliqué pour résoudre deux problèmes majeurs dans le web :

I.D.1.b.i Le filtrage de spam

La lutte contre le spam est extrêmement féroce et les nuisances apportées par ce phénomène ne se limitent pas seulement à l'afflux des emails indésirables ou la perte des emails légitimes ; simplement, nous pouvons identifier différentes sortes de spam, comme l'arnaque nigériane, virus et Phishing. Malheureusement, les techniques classiques du filtrage

[4] BjørnKirkerud en 1993 a dit que « Le défi ne consiste pas comment retrouver des informations utiles, mais plutôt comment jeter tout l'inutile »

de spam sont face à beaucoup d'inconvénients en termes de : qualité de filtrage, temps de calcul, et la qualité de services fournis aux utilisateurs.

I.D.1.b.ii Détection de plagiat

Le plagiat est un problème grave et croissant sur le Web. A tout moment, des personnes dans le monde peuvent copier votre contenu en ligne et instantanément le coller sur leurs propres travaux. Le Web est construit sur le travail acharné des gens honnêtes qui consacrent leur temps et leur énergie à créer un contenu original. L'épidémie mondiale de vol de contenu viole les droits des personnes et décourage la création de nouveaux contenus web. C'est pour cette raison, l'urgence de développer des outils efficaces de détection de plagiat est devenue une nécessité primordiale. Malheureusement, les outils de détection de plagiat automatique disponibles sur le web sont incapables de détecter les différents types de plagiat comme le plagiat des idées, plagiat avec traduction, plagiat avec synonymie, l'auto plagiat et le plagiat paraphraser.

I.D.1.c. Recherche multimédia

L'indexation et la recherche multimédia se réfère à des techniques développées pour accéder à l'image, la vidéo et les bases de données sonores sans descriptions textuelles.

I.D.1.c.i Détection des personnes suspectes

Les systèmes de vidéosurveillance sont très importants dans notre vie quotidienne. Cette technologie existe dans les aéroports, les banques, les bureaux et les maisons pour nous garder en sécurité. Il n'est pas étonnant que les caméras de sécurité dans le web deviennent la solution de surveillance préférée parmi les entreprises et les propriétaires. La demande accrue pour les appareils Smartphone, PC de poche, et les ordinateurs portables a été une force motrice derrière la surveillance vidéo à travers le web. Autrement dit, les gens veulent accéder à leurs caméras de sécurité en ligne afin qu'ils puissent aller sur leurs lieux de travail quotidiennement, sans avoir à se soucier que leur propriété est vulnérable au vol et les cambriolages. Les gens veulent conserver un œil sur leurs biens, alors qu'ils sont en déplacement, ou en vacances. Pour cela, il est nécessaire d'installer un système de vidéosurveillance accessible à travers un service web et internet. Simplement, les systèmes classiques de vidéosurveillance sont face à beaucoup de limites en termes de violation de la vie privée des personnes et de l'incapacité à identifier les personnes malveillantes qui cachent leurs visages ou se déguisent.

I.E. Conclusion

Dans ce premier chapitre, on a introduit quelques définitions de la recherche d'information qui est un sujet d'actualité avec la présence de cette énorme quantité d'information qui ne cesse d'accroitre jour après jour d'où la nécessité de concevoir des systèmes de recherche d'information performant qui répondent aux besoins des utilisateurs.

Par ailleurs, on a trouvé utile de définir quelques concepts de base liés au terme RI. Le processus de recherche d'information a pris part dans ce présent chapitre ainsi que l'évaluation des systèmes de RI en terme de performance et de rapidité.

Deuxième chapitre
Métaheuristiques et bio-inspiration

II. Métaheuristiques et bio-inspiration

II.A. Introduction

L'approche biomimétique examine les lois, les stratégies et les principes employés par les organismes vivants afin de les imiter ou de s'inspirer de ceux-ci pour répondre aux problématiques contemporaines (Benyus, 1997). La structure de la réflexion biomimétique se divise en quatre phases : Définition des champs d'application, recherche, conception puis évaluation et analyse. La première phase permet de camper le problème. Le contexte y est défini puis l'identification de la fonction biologique et des principes applicables à la problématique sont dégagés. Ensuite, la seconde phase correspond à la recherche et l'exploration des différents modèles naturels pouvant offrir une solution au problème afin d'isoler les stratégies potentiellement applicables. Puis, la troisième partie relève du processus créatif où différentes solutions sont développées à partir des stratégies précédemment dégagées. Enfin, la dernière partie porte sur l'évaluation de l'efficacité des solutions établies à partir des principes de vie applicables [24].

Ces approches sont destinées à résoudre plusieurs problèmes complexes que le modèle mathématique n'a pas aboutis à les résoudre en un temps raisonnable.

II.B. Optimisation combinatoire

L'optimisation combinatoire est un outil indispensable combinant diverses techniques des mathématiques discrètes et de l'informatique afin de résoudre des problèmes de la vie réelle [37]. Les problèmes peuvent être combinatoires (discrets) ou à variables continues, avec un seul ou plusieurs objectifs (optimisation mono-objective ou multi-objective), statiques ou dynamiques [5]. Il faut noter que, l'optimisation d'un problème multi-objectif est souvent plus compliqué et plus difficile par rapport à l'optimisation des problèmes mono-objectifs.

D'une manière simple, résoudre un problème d'optimisation combinatoire consiste à trouver l'optimum d'une fonction objectif dans le but de trouver une solution optimale dans un temps d'exécution raisonnable parmi un nombre fini de choix, souvent très grand sous certaines contraintes [38].

Néanmoins, ce but est loin d'être réalisable pour plusieurs problèmes vu leurs complexités grandissantes. La théorie de complexité présentée par Gary et son équipe de recherche [25] permet de classer les problèmes d'optimisation par rang de leurs complexités et fournit des informations pertinentes qui servent au choix de méthodes de résolution.

Les problèmes d'optimisation combinatoire sont généralement faciles à définir, mais ils sont difficiles à résoudre [8]. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent pas encore de solutions algorithmiques efficaces [18]. Ces dernières années ont été marquées par une croissance très rapide des travaux qui utilisent les méthodes d'optimisation.

II.C. Les métaheuristiques

Une Métaheuristique est l'extension d'une heuristique ce qui la rends plus complète d'une part et plus complexe d'une autre part, le but de l'extension est l'obtention d'une solution de qualité supérieure, elle est caractérisée par son comportement stochastique itératif qui vise à progresser vers un optimum global quel que soit le point de départ, et s'inspire essentiellement des systèmes naturels. La plupart des Métaheuristiques utilisent des processus

aléatoires et itératifs pour rassembler de l'information, explorer l'espace de recherche et faire face à des problèmes comme l'explosion combinatoire.

Toute Métaheuristique doit trouver un point d'équilibrage entre les deux principes : intensification et diversification où :

- L'intensification est le fait de focaliser l'effort de recherche vers les zones les plus prometteuses de l'espace de solutions ;

- La diversification consiste à répartir l'effort de recherche sur tout l'espace de recherche afin de découvrir de nouvelles zones aussi prometteuses contenant de meilleures combinaisons.

Ce compromis permet d'échapper à une convergence prématurée, le réglage de ce compromis diffère d'une Métaheuristique à une autre, en général il se fait par la modification des paramètres, dont la Métaheuristique dépend.

En général, l'intensification de la recherche, en l'incitant à explorer les combinaisons proches des meilleures combinaisons trouvées, engendre une convergence rapide. Cependant, une intensification excessive risque de faire stagner les solutions l'algorithme autour des optimums locaux. La diversification se fait généralement en introduisant une part d'aléatoire, par exemple autoriser avec une faible probabilité la recherche à choisir des voisins de moins bonne qualité.

L'équilibre entre intensification et diversification dépend du temps de calcul dont on dispose pour résoudre un problème donné. Plus ce temps est petit et plus on a intérêt à favoriser l'intensification pour converger rapidement. Cet équilibre dépend également de l'instance du problème à résoudre, plus particulièrement de la topologie de répartition des solutions réalisables. Toutefois, il est difficile de trouver les valeurs adéquates permettant d'intensifier et de diversifier la recherche à la fois [2].

Nous pouvons partager les méthodes heuristiques en deux catégories :

- Les méthodes locales : ce sont les méthodes qui convergent vers un optimum local. Elles partent d'une solution initiale qu'elles améliorent d'une manière itérative, le processus s'arrête lorsqu'une amélioration d'une solution courante est impossible ou après avoir atteint le nombre maximal d'itérations fixé au départ [8] à titre d'exemple nous citant l'algorithme les plus connus en l'occurrence le recuit simulé développé par le chercheur KirkPatrick et son équipe en 1983 [35], et la recherche tabou développé par l'équipe de recherche dirigée par Glover en 1997 [13];
- Les méthodes globales : elles ont pour objectif d'atteindre un ou plusieurs optimums globaux. Celles-ci sont d'une grande diversité : parmi elles on retrouve notamment les algorithmes génétiques [9], les algorithmes à évolution différentielle [30], la recherche dispersée [7].

Ces dernières années, une nouvelle « pseudo-classe » d'algorithmes est en train d'émerger, elle contient des méthodes hybrides. Le principe consiste à combiner des algorithmes exacts et/ou des algorithmes approchés, afin de tirer profit des points forts de chaque approche et d'améliorer le comportement global.

II.C.1. Approches constructives

Les méthodes constructives produisent des solutions admissibles en partant d'une solution initiale vide et en insérant, à chaque étape, une composante dans la solution partielle courante. Cette décision n'est jamais remise en question par la suite.

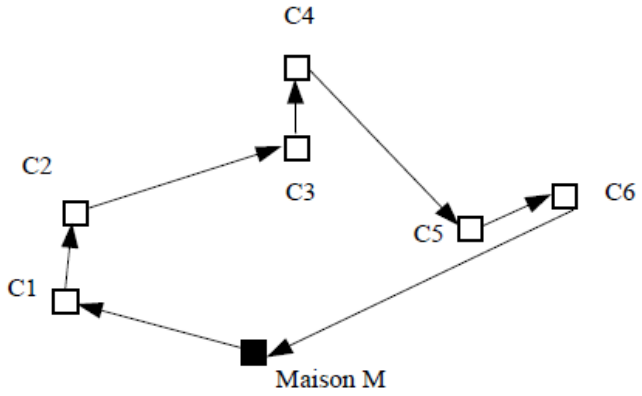

Figure II.1 – La tournée du voyageur de commerce

Pour illustrer ce type de méthodes, il suffit d'imaginer un voyageur de commerce qui doit rendre visite à un ensemble de n clients. Il peut construire sa tournée de la manière suivante : en partant de chez lui, il va chez le client le plus proche (disons C1). En quittant C1, il va chez le client le proche de C1 qu'il n'a pas encore rencontré, en ainsi de suite jusqu'à qu'il ait rendu visite à tous ces clients. En quittant le dernier client (disons Cn), il rentre chez lui. Il a ainsi construit la tournée " M - C1 - C2 - … - Cn - M " (*Figure II.1*).

Le type de recherche qui est à la base d'une méthode constructive est représenté dans la *Figure II.2*. L'idée consiste à diminuer la taille du problème à chaque étape, ce qui revient à se restreindre à un sous-ensemble X^k inclus dans X toujours plus petit. Une méthode constructive trouve une solution optimale lorsque chacun des sous-ensembles considérés contient au moins une solution optimale $s^* \in X$. Malheureusement, rares sont les cas où une telle condition est remplie avec certitude.

La majorité des méthodes constructives sont de type "glouton". A chaque étape, la solution courante est complétée de la meilleure façon possible sans tenir compte de toutes les conséquences que cela entraîne au niveau du coût de la solution finale. Dans ce sens, les méthodes de type glouton sont souvent considérées comme myopes.

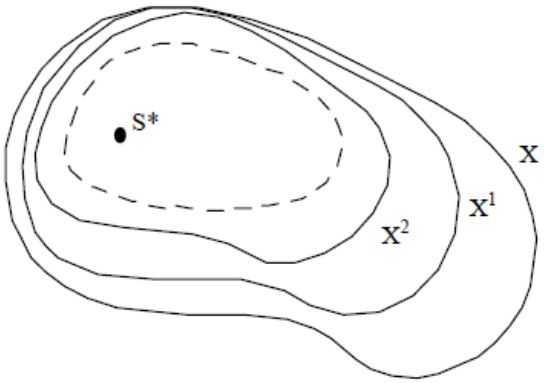

Figure II.2 – Exploration de X par approche constructive

Les méthodes constructives se distinguent par leur rapidité et leur grande simplicité. On obtient en effet très rapidement une solution admissible pour un problème donné sans avoir recours à des techniques hautement sophistiquées. Le principal défaut de ces méthodes réside malheureusement dans la qualité des solutions obtenues. Le fait de vouloir opérer à tout prix le meilleur choix à chaque étape est une stratégie dont les effets peuvent être catastrophiques à long terme. D'un point de vue théorique, l'obtention d'une solution optimale est assurée uniquement pour les problèmes qui admettent une formulation en termes de matroïdes (Gondran et Minoux, 1985).

Il est donc judicieux, dans le cas général, de mettre au point des procédures anticipant les effets secondaires et les conséquences futures occasionnées par les décisions prises lors de la construction d'une solution admissible.

II.C.2. Approches de recherche locale

Les méthodes de recherche locale sont des algorithmes itératifs qui explorent l'espace X en se déplaçant pas à pas d'une solution à une autre. Une méthode de ce type débute à partir d'une solution $s_0 \in X$ choisie arbitrairement ou alors obtenue par le biais d'une méthode constructive.

Le passage d'une solution admissible à une autre se fait sur la base d'un ensemble de modifications élémentaires qu'il s'agit de définir de cas en cas. Une solution s'obtient à partir de s en appliquant une modification élémentaire.

Le voisinage N(s) d'une solution $s \in X$ est défini comme l'ensemble des solutions admissibles atteignables depuis s en effectuant une modification élémentaire.

Un tel processus d'exploration est interrompu lorsqu'un ou plusieurs critères d'arrêt sont satisfaits. Le fonctionnement d'une méthode de recherche locale est illustré de manière générale dans la *Figure II.3*. Les passages successifs d'une solution à une solution voisine définissent un chemin au travers de l'espace des solutions admissibles.

La modélisation d'un problème d'optimisation et le choix du voisinage doivent être effectués de telle sorte qu'il existe au moins un " chemin " entre chaque solution $s \in X$ et une solution optimale s*. En effet, l'existence de tels chemins permet à la méthode de recherche locale d'atteindre une solution optimale à partir de n'importe quelle solution admissible.

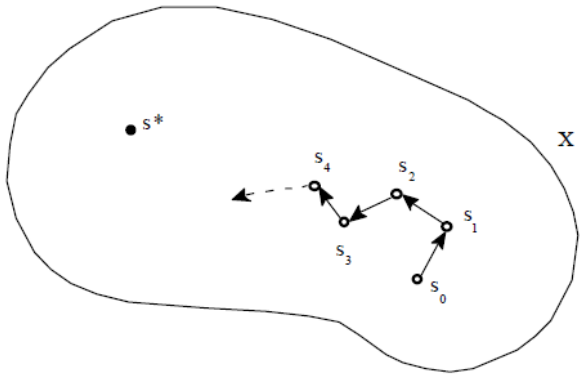

Figure II.3 – Exploration de X une approche de recherche locale

A titre d'exemple, le cas du voyageur de commerce, qui a obtenu une tournée initiale par une méthode constructive, peut essayer d'améliorer cette dernière grâce à une méthode de

recherche locale. Il lui suffit de définir comme modification élémentaire la permutation de deux clients dans sa tournée (*Figure II.4* : on rend visite d'abord à C6, puis à C5).

La méthode de descente décrite de manière générique dans l'algorithme 1 est un exemple de méthode de recherche locale. Une telle méthode progresse au travers de X en choisissant à chaque étape la meilleure solution voisine de la solution courante. Ce procédé est répété aussi longtemps que la valeur de la fonction objectif diminue. La recherche s'interrompt dès lors qu'un minimum local de f est atteint.

Historiquement, les méthodes de descente ont toujours compté parmi les méthodes heuristiques les plus populaires pour traiter les problèmes d'optimisation combinatoire. Toutefois elles comportent deux obstacles majeurs qui limitent considérablement leur efficacité:

- suivant la taille et la structure du voisinage N(s) considéré, la recherche de la meilleure solution voisine est un problème qui peut être aussi difficile que le problème (P) initial ;

- une méthode de descente est incapable de progresser au-delà du premier minimum local rencontré. Or les problèmes d'optimisation combinatoire comportent typiquement de nombreux optima locaux pour lesquels la valeur de la fonction objectif peut être fort éloignée de la valeur optimale.

Initialisation

choisir une solution admissible initiale s∈X ;

poser s*:=s ;

Processus itératif

tant que le critère d'arrêt n'est pas satisfait faire

générer N(s) ;

déterminer s'∈N(s) telle que

$$f(s') = \min_{s''\in N(s)} f(s'')$$

s := s' ;

si f(s) < f(s*) alors s*:=s ;

sinon le critère d'arrêt est satisfait

Algorithme II.1 – La méthode de descente

Pour faire face à ces carences, des méthodes de recherche locale plus sophistiquées ont été développées au cours de ces vingt dernières années. Ces méthodes acceptent des solutions voisines moins bonnes que la solution courante afin d'échapper aux minima locaux de la fonction f. En règle générale, seule une portion du voisinage courant est explorée à chaque étape. Les méthodes les plus connues seront présentées dans les paragraphes suivants. Les différences principales entre ces méthodes se situent au niveau du choix de la solution voisine et au niveau du critère d'arrêt. La recherche peut être clairement interrompue lorsqu'une solution suffisamment proche de la solution optimale est atteinte. Malheureusement rares sont les problèmes difficiles où la valeur de la solution optimale est connue.

Les méthodes présentées ci-dessous sont en général beaucoup plus performantes qu'une simple méthode de descente mais également beaucoup plus coûteuses en termes de ressources informatiques. Leur mise en œuvre doit généralement tenir compte du temps de réponse maximal autorisé par l'utilisateur du programme. Il convient de signaler pour conclure qu'un effort non négligeable est nécessaire pour ajuster convenablement les paramètres qu'elles font intervenir dans le but de guider efficacement la recherche au travers de l'ensemble X.

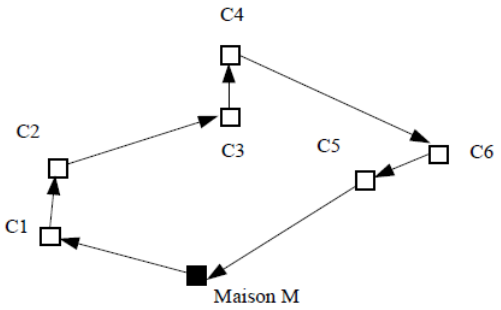

Figure II.4 – La tournée modifiée par une permutation

II.C.2.a. Le recuit simulé

Les origines de la méthode du recuit simulé remontent aux expériences de Metropolis et al. (Metropolis et al, 1953). Leurs travaux ont abouti à un algorithme simple pour simuler l'évolution d'un système physique instable vers un état d'équilibre thermique à une température t fixée. L'état du système physique est caractérisé par la position exacte de l'ensemble des atomes qui le composent. Metropolis et al. utilisent une méthode de Monte Carlo pour générer une suite d'états successifs du système en partant d'un état initial donné. Tout nouvel état est obtenu en faisant subir un déplacement infinitésimal aléatoire à un atome quelconque. Soit ΔE la différence d'énergie occasionnée par une telle perturbation. Le nouvel état est accepté si l'énergie du système diminue (ΔE < 0). Dans le cas contraire (ΔE = 0), il est accepté avec une certaine probabilité :

$$\text{prob}(\Delta E, t) = \exp\left(\frac{-\Delta E}{k_B . t}\right)$$

où t est la température du système et kB une constante physique connue sous le nom de constante de Boltzmann.

A chaque étape, l'acceptation d'un nouvel état dont l'énergie n'est pas inférieure à celle de l'état courant est décidée en générant de manière aléatoire un nombre q ∈ [0,1 [. Si q est inférieur ou égal à prob(ΔE,t), alors le nouvel état est accepté. Autrement l'état courant est maintenu. Metropolis et al. ont montré que l'utilisation répétée d'une telle règle fait évoluer le système vers un état d'équilibre thermique.

Le recuit simulé est une méthode de recherche locale dont le mécanisme de recherche est calqué sur l'algorithme de Metropolis et al. et les principes du recuit thermodynamique. L'idée consiste à utiliser l'algorithme de Metropolis et al. pour des valeurs décroissantes de la température t. Le refroidissement progressif d'un système de particules est simulé en faisant une analogie entre l'énergie du système et la fonction objectif du problème (P) d'une part, et entre les états du système et les solutions admissibles de (P) d'autre part. Pour atteindre des états avec une énergie aussi faible que possible, on porte initialement le système à très haute température puis on le refroidit petit à petit. Lorsque la température diminue, les mouvements

d'atomes deviennent de moins en moins aléatoires et le système aura tendance à se trouver dans des états à basse énergie. Le refroidissement du système doit se faire très lentement pour avoir l'assurance d'atteindre un état d'équilibre à chaque température t. Lorsqu'aucun état nouveau n'est accepté à une température t donnée, on considère que le système est gelé et on suppose qu'il a atteint un niveau d'énergie minimum.

Kirkpatrick et al. (Kirkpatrick et al., 1983) et Cerny (Cerny, 1985) ont été les premiers à s'inspirer d'une telle technique pour résoudre des problèmes d'optimisation combinatoire. Le voisinage N(s) d'une solution s∈X s'apparente à l'ensemble des états atteignables depuis l'état courant en faisant subir des déplacements infinitésimaux aux atomes du système physique. A chaque itération, une seule solution voisine s'est générée. Celle-ci est acceptée si elle est meilleure que la solution courante s. Dans le cas contraire, on procède comme dans l'algorithme de Metropolis et al. et la nouvelle solution s est acceptée avec une certaine probabilité prob(Δf,t) qui dépend de l'importance de la détérioration Δf=f(s')-f(s) et d'un paramètre t correspondant à la température. Les changements de température sont effectués sur la base d'un schéma de refroidissement précis. En règle générale, la température est diminuée par paliers à chaque fois qu'un certain nombre d'itérations est effectué. La meilleure solution trouvée est mémorisée dans la variable s*. L'algorithme est interrompu lorsqu'aucune solution voisine n'a été acceptée pendant un cycle complet d'itérations à température constante.

La performance du recuit simulé est étroitement liée au schéma de refroidissement considéré. De nombreuses études théoriques ont été effectuées à ce sujet et plusieurs variantes ont été proposées (Collins et al, 1988), (Osman et Christofides, 1994). Notons également qu'une revue détaillée de la littérature a été effectuée par Collins et al. (Collins et al, 1988).

II.C.2.b. La recherche tabou

La technique tabou est une méthode itérative générale d'optimisation combinatoire qui a été introduite par Glover (Glover, 1986).

Comme indiqué précédemment, le déplacement d'une solution courante s vers une solution voisine s' est choisi de telle sorte que

$$f(s') = \min_{s''\in N(s)} f(s'')$$

Tant que l'on ne se trouve pas dans un optimum local, toute méthode itérative se comporte donc comme la méthode de descente et améliore à chaque étape la valeur de la fonction objectif. Lorsque l'on atteint par contre un optimum local, la règle de déplacement donnée ci-dessus permet de choisir le moins mauvais des voisins, c'est-à-dire celui qui donne un accroissement aussi faible que possible de la fonction objectif.

L'inconvénient que représenterait une méthode basée sur cet unique principe est que si un minimum local s se trouve au fond d'une vallée profonde, il sera impossible de ressortir de celle-ci en une seule itération, et un déplacement de la solution s vers une solution s'∈N(s) avec f(s')>f(s) peut provoquer le déplacement inverse à l'itération suivante, puisqu'en général s∈N(s') et f(s)<f(s') ; on risque ainsi de "cycler" autour de ce minimum local.

C'est pour cette raison que la méthode tabou s'appuie sur un deuxième principe qui consiste à garder en mémoire les dernières solutions visitées et à interdire le retour vers celles-ci pour un nombre fixé d'itérations, le but étant de donner assez de temps à l'algorithme pour lui permettre de sortir d'un minimum local. En d'autres termes, la méthode tabou conserve à

chaque étape une liste T de solutions "taboues", vers lesquelles il est interdit de se déplacer momentanément. L'espace nécessaire pour enregistrer un ensemble de solutions taboues peut s'avérer important en place mémoire. Pour cette raison, il est parfois préférable d'interdire uniquement un ensemble de mouvements qui ramèneraient à une solution déjà visitée. Ces mouvements interdits sont appelés mouvements tabous.

Lors du choix de la meilleure solution s'∈ N(s), il est possible que l'on ait à départager plusieurs candidats donnant certes une même valeur à la fonction objectif, mais ne nous dirigeant pas tous vers un optimum global. Il est ainsi parfois souhaitable de pouvoir retourner vers une solution visitée s, malgré le fait qu'elle fasse partie de la liste T des solutions taboues, ceci afin d'explorer une nouvelle région voisine de s. Pour cette raison, la méthode tabou fait intervenir un nouvel ingrédient appelé fonction d'aspiration et défini sur toutes les valeurs de la fonction objectif : lorsqu'une solution s' voisine de la solution s fait partie de T et satisfait de plus l'aspiration (c'est-à-dire f(s')<A(f(s))), on lève le statut tabou de cette solution s' et elle devient donc candidate lors de la sélection du meilleur voisin de s. En général, A(f(s)) prend la valeur de la meilleure solution s* rencontrée (on “aspire” donc à déterminer une solution meilleure que s*).

Pour certains problèmes, le voisinage N(s) de la solution courante s est de grande taille et de surcroît le seul moyen de déterminer la solution s' minimisant f sur N(s) est de passer en revue l'ensemble N(s) tout entier; on préfère alors générer un sous-ensemble N' inclus dans N(s) ne contenant qu'un échantillon de solutions voisines à s et on choisit la solution s'∈N' de valeur f(s') minimale.

Il faut encore définir une condition d'arrêt. On se donne en général un nombre maximum nbmax d'itérations entre deux améliorations de la meilleure solution s* rencontrée. Dans certains cas, il est possible de déterminer une borne inférieure f de la fonction objectif et on peut alors stopper la recherche lorsqu'on a atteint une solution s de valeur f(s) proche de f.

Plusieurs stratégies ont été proposées récemment afin d’améliorer l’efficacité de la méthode tabou présenté (Glover, 1997). L’intensification et la diversification de la recherche sont deux d’entre elles. L’intensification consiste à explorer en détail une région de X jugée prometteuse. Sa mise en œuvre réside le plus souvent en un élargissement temporaire du voisinage de la solution courante dans le but de visiter un ensemble de solutions partageant certaines propriétés. La diversification est une technique complémentaire à l’intensification. Son objectif est de diriger la procédure de recherche vers des régions inexplorées de l’espace X. La stratégie de diversification la plus simple consiste à redémarrer périodiquement le processus de recherche à partir d’une solution générée aléatoirement ou choisie judicieusement dans une région non encore visitée de l’ensemble des solutions admissibles.

Le lecteur intéressé trouvera une description plus fournie de la méthode tabou ainsi qu’un exemple détaillé dans (Widmer et al., 2001).

II.C.3. Approches évolutionnaires

Les sciences de la vie et les processus naturels ont de tout temps fasciné les ingénieurs. Ces derniers n’hésitent pas à s’inspirer des structures et des mécanismes du monde vivant pour développer des objets artificiels utilisables dans des contextes variés. Dans le domaine de l’optimisation combinatoire, la complexité des phénomènes naturels a servi de modèle pour des algorithmes toujours plus sophistiqués ces vingt-cinq dernières années. Les

méthodes évolutionnaires qui sont présentées dans ce paragraphe constituent la base d'un nouveau champ de la programmation informatique en pleine effervescence.

Contrairement aux méthodes constructives et de recherche locale qui font intervenir une solution unique (partielle ou non), les méthodes évolutionnaires manipulent un groupe de solutions admissibles à chacune des étapes du processus de recherche. L'idée centrale consiste à utiliser régulièrement les propriétés collectives d'un ensemble de solutions distinguables, appelé population, dans le but de guider efficacement la recherche vers de bonnes solutions dans l'espace X. En règle générale, la taille de la population reste constante tout au long du processus. Après avoir généré une population initiale de solutions, aléatoirement ou par l'intermédiaire d'une méthode constructive, une méthode évolutive tente d'améliorer la qualité moyenne de la population courante en ayant recours à des principes d'évolution naturelle. Dans notre terminologie, le processus cyclique qui est à la base d'une méthode évolutive est composé d'une phase de coopération et d'une phase d'adaptation individuelle qui se succèdent à tour de rôle. Ce formalisme nouveau s'applique à la plupart des méthodes évolutionnaires développées à ce jour.

Dans la phase de coopération, les solutions de la population courante sont comparées puis combinées entre elles dans le but de produire des solutions inédites et de bonne qualité à long terme. L'échange d'information qui en résulte se traduit par l'apparition de nouvelles solutions admissibles qui héritent des caractéristiques prédominantes contenues dans les solutions de la population courante. Dans la phase d'adaptation individuelle, les solutions évoluent de manière indépendante en respectant un ensemble de règles prédéfini. Les modifications subies par chacune d'entre elles se font sans aucune interaction avec les autres solutions de la population. Une nouvelle génération de solutions est créée au terme de chaque phase d'adaptation individuelle.

Le mécanisme de recherche qui est à la base d'une approche évolutive est représenté sommairement dans la *Figure II.5*. Le but est de repérer des solutions aussi bonnes que possible en manipulant à chaque étape un ensemble de solutions localisées dans différentes régions prometteuses de l'espace X.

Par la suite, on dira qu'une méthode évolutive converge prématurément ou traverse une crise de diversité lorsque la population courante contient un pourcentage élevé de solutions identiques. Différents mécanismes peuvent être incorporés pour pallier cet inconvénient, en prenant des mesures afin de réintroduire une diversité d'information suffisante au sein de la population courante.

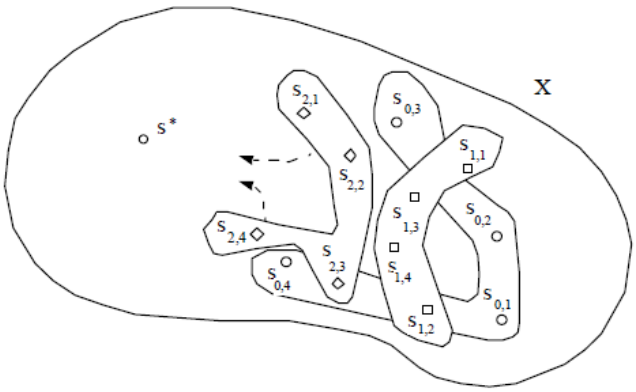

Figure II.5 – Exploration de X par une approche évolutive

II.C.3.a. Algorithmes génétiques

Les algorithmes génétiques sont des méthodes évolutionnaires qui s'inspirent fortement des mécanismes biologiques liés aux principes de sélection et d'évolution naturelle. Développés initialement par Holland (Holland, 1975) pour répondre à des besoins spécifiques en biologie, les algorithmes génétiques ont rapidement été adaptés à des contextes très variés. Dans un algorithme génétique simple (Goldberg, 1989), la recherche est réglée par trois opérateurs qui sont appliqués successivement. La phase de coopération est gouvernée par un opérateur de reproduction et un opérateur de combinaison (ou "crossover") alors que la phase d'adaptation individuelle fait appel à un opérateur de mutation. Il est important de souligner que les concepts qui sont à la base des algorithmes génétiques sont extrêmement simples. En effet, ils font uniquement intervenir des nombres générés aléatoirement et un ensemble de règles probabilistes très générales qui ne tiennent pas forcément compte de toutes les particularités du problème traité.

Le lecteur intéressé trouvera une excellente description très détaillée des algorithmes génétiques dans (Portmann et Vignier, 2001).

II.C.4. L'approche de l'intelligence en essaim (Swarm Intelligence)

L'intelligence en essaim (SI : Swarm Intelligence) est le fruit de plusieurs travaux de modélisation mathématique et informatique des phénomènes biologiques rencontrés en éthologie [11], toute tentative de concevoir des algorithmes ou des dispositifs de résolution des problèmes, distribués, inspirés des comportements collectifs des insectes sociaux ou d'autres sociétés animales entre dans le cadre de l'intelligence en essaim [11]. Selon Gerardo Beni et JingWang, l'intelligence en essaim est le comportement collectif décentralisé, auto-organisé des systèmes naturels ou artificiels [16].

La source d'inspiration vient souvent de la nature, en particulier les systèmes biologiques, et de l'observation du comportement des insectes sociaux, vu la richesse de leur comportement à travers les interactions et à partir de leur environnement, où les individus sont moins compliqués et plus simples [19], est le principal motif de cette modélisation.

L'ensemble d'algorithmes de l'intelligence en essaim se base sur une population d'agents simples qui interagissent localement les uns avec les autres d'une part et avec leur environnement d'autre part, sans qu'il n'y ait un contrôle centralisé qui guide le comportement des agents individuels, seules les interactions locales entre les agents conduisent à l'émergence d'un comportement collectif global auto-organisé. Notons que les

agents ont une capacité individuelle très limitée, ils peuvent conjointement effectuer un nombre fini de tâches simples nécessaires à leur survie [3].

Quatre principes gouvernant l'intelligence en essaim :

- Le feedback positif : il permet de renforcer les meilleurs choix dans le système ;
- Le feedback négatif : il permet d'ignorer et de supprimer les mauvais choix dans le système ;
- L'aspect aléatoire : il permet la bonne exploration de l'espace de solution, d'une manière indépendante de la qualité, favorisant le principe de diversification ;
- L'interaction multiple qui permet la construction des meilleures solutions et choix.

II.C.4.a. Colonies de fourmis

Les performances collectives des insectes sociaux, tels que les fourmis, les abeilles, les guêpes ou les termites, intriguent les entomologistes depuis de nombreuses années. L'interrogation majeure concerne les mécanismes qui permettent aux individus d'une même colonie de régler leurs activités et de favoriser la survie de l'espèce. Tout se passe comme si un agent invisible, au centre de la colonie, coordonnait les activités de tous les individus. Des études ont montré que ce comportement global résultait d'une multitude d'interactions locales particulièrement simples. La nature de ces interactions, les mécanismes de traitement de l'information et la différence entre le comportement solitaire et le comportement social sont restés longtemps mystérieux. Lors de la réalisation d'une tâche spécifique par une colonie d'insectes, il a été observé récemment que la coordination des travaux ne dépendait pas directement des ouvriers mais plutôt de l'état d'avancement de la tâche. L'ouvrier ne dirige pas son travail ; il est guidé par lui. Tout insecte, en travaillant, modifie la forme de la stimulation qui déclenche son comportement et provoque ainsi l'apparition d'une nouvelle stimulation qui déclenchera d'autres réactions chez lui-même ou chez un de ses congénères.

Pour illustrer l'apparition de structures collectives dans une société d'insectes, il convient de citer l'exemple d'une colonie de fourmis qui est à la recherche d'une source de nourriture. Initialement, les fourmis quittent le nid et se déplacent de manière aléatoire. Lorsqu'une fourmi découvre par hasard une source de nourriture, elle informe ses congénères de sa découverte en déposant sur le sol une marque transitoire lors de son retour vers le nid. Cette marque n'est autre qu'une substance chimique, nommée phéromone, qui va guider les autres fourmis vers la même source de nourriture. A leur retour, ces dernières déposent également de la phéromone sur le sol et renforcent ainsi le marquage de la piste qui mène du nid à la source de nourriture découverte. Le renforcement du marquage par les phéromones de la piste la plus fréquentée optimise la collecte de nourriture. A long terme, les fourmis exploiteront uniquement la source la plus proche car la trace conduisant aux sources éloignées s'évaporera et deviendra indécelable. Cet exemple montre que la colonie de fourmis converge vers une solution optimale alors que chaque fourmi n'a accès qu'à une information locale et qu'elle est incapable de résoudre seule le problème dans un délai raisonnable.

Depuis quelques années, les ingénieurs s'intéressent au comportement des insectes sociaux afin de créer une nouvelle forme de "résolution collective" de problèmes. L'algorithme de la fourmi, développé initialement par Colorni et al. (Colorni et al., 1991) est une méthode évolutive dont les mécanismes de recherche s'inspirent fortement du comportement collectif d'une colonie de fourmis. Dans la phase de coopération, chaque solution de la population courante est examinée à tour de rôle dans le but de mettre à jour une mémoire globale. La phase d'adaptation individuelle fait intervenir ensuite une méthode

constructive qui utilise l'information contenue dans la mémoire globale pour créer des nouvelles solutions admissibles. Une telle approche utilise de manière répétée une méthode constructive en faisant intervenir à chaque fois l'expérience accumulée lors des précédentes applications de la méthode. Chaque application de la méthode constructive correspond au travail réalisé par une fourmi isolée.

II.C.5. Approches hybrides

Les méthodes évolutionnaires, et plus particulièrement les algorithmes génétiques, ont été largement étudiés depuis les tous premiers développements réalisés au début des années 70 (Holland, 1975). Les nombreuses adaptations qui ont été proposées dans la littérature comblent les déficiences principales des méthodes évolutionnaires classiques dont les performances globales sont souvent bien inférieures à celle d'une méthode de recherche locale telle que la méthode taboue ou le recuit simulé. Dans un cadre plus spécifique, il est désormais établi qu'un algorithme génétique simple n'est pas en mesure de fournir de bons résultats lorsque l'espace des solutions est très contraint. Grefenstette (Greffenstette, 1987) a montré qu'il était possible de tirer profit des particularités du problème étudié dans la définition de tous les opérateurs qui composent un algorithme génétique.

La plupart des innovations introduites dans le domaine des méthodes évolutionnaires fait appel à des concepts qui ne sont plus liés à des principes d'évolution naturelle. Des résultats fort intéressants ont été obtenus récemment en insérant une méthode de recherche locale dans la phase d'adaptation individuelle d'une méthode évolutive. Dans ce qui suit, référence est faite à cette nouvelle méthode de recherche combinée en termes d'algorithme hybride.

La force d'un algorithme hybride réside dans la combinaison de deux principes de recherche fondamentalement différents. Le rôle de la méthode de recherche locale est d'explorer en profondeur une région donnée de X alors que la méthode évolutive introduit des règles de conduite générales dans le but de guider la recherche au travers de X. Dans ce sens, les opérateurs de combinaison ont un effet diversificateur bénéfique à long terme. A notre connaissance, les origines des algorithmes hybrides remontent aux travaux de Glover (Glover, 1977), refenstette

(Greffenstette, 1987) et Mühlenbein et al. (Mühlenbein et al., 1988). Chacun a fait intervenir une méthode de descente simple (*algorithme II.1*) pour accroître la performance d'une méthode évolutive existante. Glover a utilisé une méthode de recherche distribuée alors que Grefenstette et Mühlenbein et al. ont eu recours à un algorithme génétique pour résoudre respectivement des problèmes de programmation en nombres entiers et de voyageur de commerce. Le recuit simulé et la méthode tabou sont des améliorations de la méthode de descente simple. Il est donc naturel de les utiliser au sein d'une méthode évolutive pour accroître davantage la performance de cette dernière.

Dans sa thèse, Costa propose deux algorithmes hybrides basés sur les deux schémas de combinaison décrits dans la *Figure II.6* ci-dessous (Costa, 1995). L'algorithme qui en découle est appelé algorithme tabou évolutif ou algorithme de descente évolutif selon le type de méthode de recherche locale utilisé. Dans le premier cas, la méthode de recherche locale joue le rôle de l'opérateur de mutation alors que dans le second elle remplace l'opérateur de reproduction. Celui-ci a été supprimé pour réduire les risques de convergence prématurée qui sont souvent élevés dans un algorithme hybride dont la méthode de recherche locale est déterministe.

L'algorithme tabou évolutif y est appliqué à des problèmes de confection d'horaires sportifs. Il montre que la performance de l'algorithme obtenu est significativement meilleure que celle de la méthode taboue exécutée séparément durant un intervalle de temps comparable. Un tel comportement résulte de la complémentarité existant entre la méthode taboue et les algorithmes génétiques. La méthode taboue nécessite un effort important de modélisation et d'ajustement de paramètres pour qu'elle soit réellement efficace. A l'inverse, les algorithmes génétiques sont des méthodes qui sont certes moins efficaces en tant que telles, mais qui ont le grand avantage d'être robustes et basées sur des règles extrêmement simples. De plus, Costa présente une adaptation de l'algorithme de descente évolutif pour résoudre des problèmes de coloration dans les graphes (Costa et al., 1995). La performance de l'algorithme obtenu est bien supérieure à celle de l'algorithme génétique et de la méthode de recherche locale auxquels il fait appel.

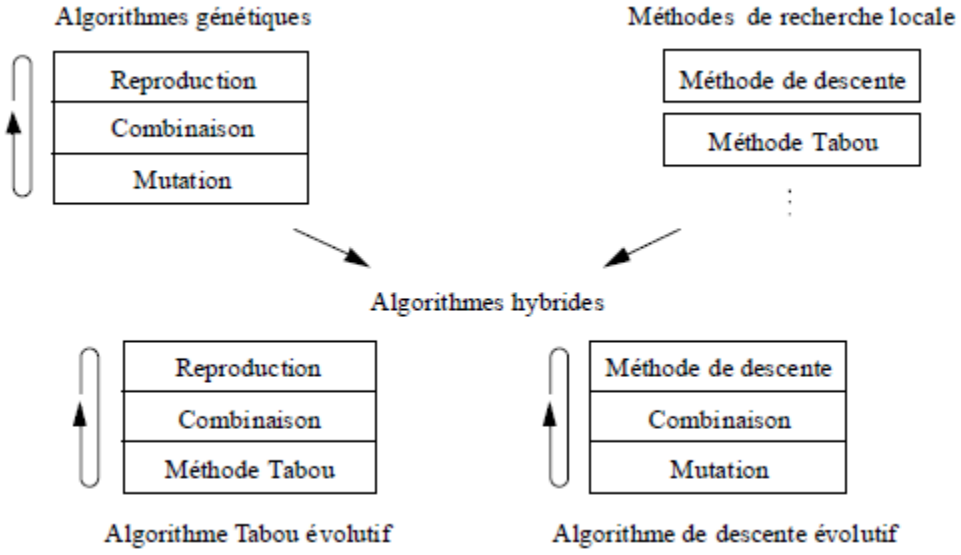

Figure II.6 – Deux schémas de combinaison utilisant un algorithme génétique

Les résultats obtenus à l'aide d'un algorithme hybride sont habituellement de très bonne qualité. Malheureusement, les temps de calcul nécessaires pour atteindre une solution de qualité donnée peuvent devenir prohibitifs. Après avoir comparé plusieurs approches pour résoudre un éventail de problèmes d'affectation quadratique, Taillard (Taillard, 1994) conclut que les algorithmes hybrides sont parmi les plus puissants mais également les plus coûteux en temps. Le choix de la méthode à utiliser dépend donc en grande partie du temps de traitement disponible pour résoudre un problème particulier. Les recherches futures devraient s'orienter vers une parallélisations des algorithmes hybrides dans le but de réduire le temps de calcul et de permettre ainsi la résolution de problèmes de plus grande taille.

II.D. Conclusion

Le but d'une Métaheuristique est l'obtention d'une solution de qualité supérieure, elle est caractérisée par son comportement stochastique itératif qui vise à progresser vers un optimum global quel que soit le point de départ, et s'inspire essentiellement des systèmes naturels. La plupart des métaheuristiques utilisent des processus aléatoires et itératifs pour explorer l'espace de recherche et faire face à des problèmes comme l'explosion combinatoire.

Dance ce chapitre, on a présenté un état de l'art sur les algorithmes et les approches qui existent dans la littérature tt qui vise à trouver une solution même approchée dans un temps raisonnable. Parmi les plus connus, on a cité : les approches évolutionnaires, les approches de recherche locale, l'approche de l'intelligence en essaim et enfin les approches hybrides qui combinent plusieurs algorithmes dans le but de réduire le temps de calcul et de permettre ainsi la résolution de problèmes complexes.

Troisième chapitre

Modèle des éléphants sociaux d'Asie pour le Recherche d'Information

III. Modèle des éléphants d'Asie sociaux pour la RI

III.A. Introduction

La nature est une puissante source d'inspiration pour résoudre des problèmes informatiques complexes, car elle montre des phénomènes extrêmement divers, dynamiques, robustes, complexes et fascinants. Elle trouve toujours la solution optimale pour résoudre son problème et maintient l'équilibre parfait entre ses composants. La nouvelle ère est ouverte avec des algorithmes inspirés de la nature (bio-inspirés) qui sont des métaheuristiques imitant la nature pour résoudre les problèmes d'optimisation. Au cours des dernières décennies, de nombreux efforts de recherche ont été concentrés dans ce domaine particulier. Ce chapitre présente un aperçu des algorithmes inspirés par la nature, regroupés par domaine biologique qui a inspiré chacun d'entre eux.

La bio-inspiration est l'imitation d'un processus biologique comme les métaphores ou les phénomènes naturels pour le développement de nouveaux algorithmes. Ce champ d'étude a tricoté vaguement un ensemble des sous-domaines liés aux thèmes du connexionnisme, le comportement social et l'émergence. Il est souvent étroitement lié au domaine de l'intelligence artificielle, comme beaucoup de ses activités sont liés à l'apprentissage artificiel. Les algorithmes bio-inspirés sont avérés significativement plus robustes et adaptables que les algorithmes traditionnels. Les notions de robustesse, d'émergence, d'auto-organisation d'adaptabilité, réactivité et de distribution sont donc sous-jacentes dans ces algorithmes et font même partie de leurs fondements. La première étape dans la construction d'un algorithme bio-inspiré est de construire des composants imitant le comportement de leurs homologues biologiques. Ces composants essaient ensuite d'atteindre l'objectif global défini pour eux.

III.B. Comportement des éléphants Sociaux d'Asie

Les éléphants d'Asie sont tout à fait sociaux. Ils forment des troupeaux stables d'environ 20 individus. Ces groupes sont menés par la femelle la plus âgée (la matriarche) qui coordonne les mouvements du troupeau à **la recherche de nourriture et d'eau**.

La matriarche détient le savoir du groupe, elle connaît les routes migratoires, le rythme des saisons et les endroits importants pour trouver l'eau et la végétation.

Les troupeaux peuvent temporairement se casser en de petits sous-groupes tout en maintenant **le contact (lien d'amitié)** par des vocalisations de fond de basse fréquence.

III.C. L'origine du Classificateur des Éléphants d'Asie Sociaux [17]

Les éléphants figurent parmi les animaux les plus intelligents au monde et vivent au sein d'une société étroitement soudée. Notre idée était de produire un nouveau classificateur basé sur les résultats des études de Byrne & al dans [32] qui ont prouvés que les éléphants d'Asie au même titre que les chimpanzés et les dauphins ont une vie sociale dynamique et complexe basée sur l'esprit du groupe, une carte mentale extraordinairement précise de leur environnement, des amitiés de longue durées et des moyens de communication à long portées.

III.D. La source d'inspiration du CEAS [17]

Les éléphants d'Asie vivent dans des groupes de familles dirigés par la femelle la plus âgée et la plus expérimentée qui coordonne le mouvement du troupeau. Elle est la matriarche,

grande sœur, mère, tante, grand-mère et grand-tante pour tous les membres du groupe. La matriarche détient le savoir du groupe, elle connaît les routes migratoires, le rythme des saisons et les endroits importants pour trouver l'eau et la végétation. Les troupeaux peuvent temporairement être divisés pour rechercher les points d'eau ou la nourriture, tout en maintenant le contact [20]. La vie sociale des éléphants d'Asie est caractérisée par deux phénomènes :

III.D.1. Les liens d'amitié [17]

Les éléphants communiquent entre eux directement et discrètement jusqu'à 10 KM de distance grâce à des infrasons inaudibles pour l'être humain qui sont émis par des contractions régulières des muscles de l'organe vocal. Des fois des éléphants peuvent nous sembler solitaires, mais ils sont sans doute en communication avec les autres sans que nous le voyons ni ne l'entendions [22]. Les expériences ont montré que les éléphants ont une connaissance des identités individuelles et ils sont capables de reconnaitre les traces des autres membres de leur famille [32]. Ils rejoindront des appels de contact effectué par leurs amis (membre de leur famille ou ancien groupe) et surtout si l'éléphant appelant, a un indice d'association élevé avec le groupe. Par contre quand les éléphants entendent des appels de contact inconnu (par des éléphants avec lesquels ils n'ont pas des liens d'amitié) leur cohésion spatiale s'accroît et ils se retirent de la région.

III.D.2. Recherche de la nourriture [17]

L'organisation de la vie sociale des éléphants a un avantage pratique ; lors des périodes de sécheresse où les ressources sont rares, les liens se resserrent et des amis éloignés se rapprochent. Certains s'associent même pour assurer les meilleures places autour des points d'eau en expulsant les éléphants inconnus qui s'y trouvent. Pour les chercheurs, ce tissu de relations complexes est un véritable réseau social qui demande des capacités cognitives importantes. Les discussions à longue distance par des infrasons jouent un rôle important dans le maintien des amitiés ainsi que la mémoire de ces géants tranquilles est un outil essentiel pour se rappeler des services rendus ou des ennuis causés par les autres éléphants [21]. Chaque éléphant dans une situation de sécheresse cherche les points d'eau et lorsqu'il trouve, il envoie des signaux pour informer ses amis de l'endroit de l'eau. Les éléphants conservent des liens forts même après des absences de plus d'un an. Le scénario qui résume le phénomène social des éléphants d'Asie pour la recherche des points d'eau en cas de sécheresse est le suivant : au départ, un ensemble d'éléphants sont à la recherche d'un point d'eau dans un espace. Les éléphants ne connaissent pas l'endroit de ce point exactement, mais ils connaissent à quelle distance se trouvent et les positions de leurs amis (éléphants du même groupe). La question qui se pose : quelle est la stratégie nécessaire pour trouver le point d'eau dans des bonnes conditions ? La meilleure solution est de suivre les directives de la matriarche et les traces des éléphants ayant les meilleures positions par rapport au point d'eau avec les qu'elles, ils ont un lien d'amitié très fort.

III.E. L'application biomimétique (Passage du naturel à l'artificiel) [17]

Cette partie est dédiée au passage de la vie naturelle des éléphants d'Asie sociaux vers la vie artificielle comme la montre le tableau III.1.

Naturel	**Artificiel**
L'environnement	Espace de recherche
L'éléphant	Instance (Document)
Groupe d'éléphant	Groupe de documents (Corpus)
Matriarche (femelle la plus âgée)	Représente le document qui a le plus de score
Meilleur individu (initialisation)	Le plus proche du centroïde
Meilleur individu (en cours de traitement)	Le plus proche du point d'eau
Lien d'amitié entre l'éléphant *i* et le meilleur individu	α
Lien d'amitié entre l'éléphant *i* et la matriarche	β
Communication entre l'éléphant *i* et le meilleur individu	$\left\|ME_{\mathrm{T}}^{g} - E_{\mathrm{T}}^{i}\right\|$
Communication entre l'éléphant *i* et la matriarche	$\left\|PE_{\mathrm{T}}^{g} - E_{\mathrm{T}}^{i}\right\|$

Tableau III.1 – Glossaire des éléphants d'Asie sociaux pour la Classification Supervisée

III.F. Le processus général [17]

Chaque nouvelle instance (éléphant) est classée (rejoint le point d'eau trouvé par son groupe familiale) selon une fonction de fitness basée sur sa position précédente, sa vitesse de déplacement, la fréquence d'appel au contact et le lien d'amitié qui existe avec les membres de son groupe (surtout l'éléphant le plus proche du point d'eau et la matriarche). L'entrée du CEAS est un ensemble de vecteurs (instances), divisé en deux parties la base d'apprentissage et la base de test. Le processus général de ce classificateur est détaillé dans la *Figure III.1* [17] et les étapes de son fonctionnement sont discutées par la suite :

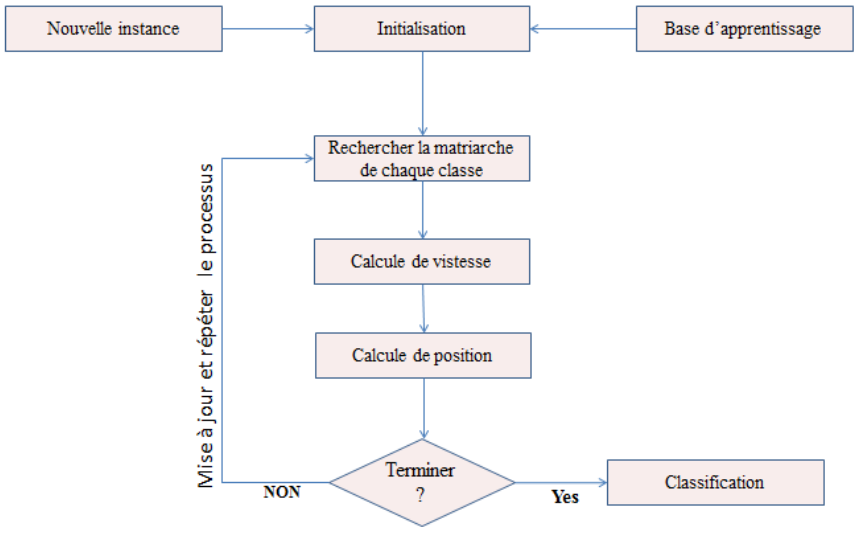

Figure III.1 – Architecture générale du CEAS

III.F.1. Initialisation [17]

Au départ, la position E_0^i et la vitesse V_0^i de chaque instance par rapport à chaque classe g sont calculées par l'intermédiaire des équations (1) et (2) :

$$\mathrm{VE}_0^i = score(i) \ldots\ldots (1)$$

$$E_0^i(g) = \textit{la distance entre l'instance i et le centroïd de la classe g} \ldots\ldots (2)$$

- $VE_0^i(g)$: La vitesse de déplacement initiale de l'instance i.
- $E_0^i(g)$: La position initiale de l'instance t par rapport à la classe g.
- *Score(i)* : La somme des poids des composants du vecteur de l'instance i.

Pour la classification d'une nouvelle instance (de la base de test) le processus suivant est lancé :

III.F.2. Matriarche [17]

Nous cherchons la matriarche de chaque classe (groupe d'éléphants) qu'est l'instance avec le plus grand score (l'éléphant femelle, la plus âgée et la plus expérimentée).

$$Mt(g) = (\max(score(i)))g$$

- $Mt(g)$: L'instance matriarche à l'instant t dans la classe g.
- $(\max(score(i)))g$: l'instance qui a le plus grand score dans la classe g.

III.F.3. La vitesse [17]

La vitesse de déplacement de chaque instance change d'un instant t à l'instant t+1 par l'équation 4.21 :

$$VE_{T+1}^i(g) = \frac{VE_T^i}{\alpha(|ME_T^g - E_T^i|) + \beta(|PE_T^g - E_T^i|)}$$

- VE_T^i : La vitesse de déplacement de l'instance i à l'instant t par rapport à la classe g.
- ME_T^g : La position du meilleur individu à l'instance t dans la classe g (initialement c'est l'individu le plus proche du barycentre de la classe).
- E_T^i : La position de l'instance i à l'instant t par rapport à la classe g.
- PE_T^g : La position de la matriarche de la classe g à l'instant t.
- α : Le lien d'amitié qui existe entre l'éléphant avec la meilleur position et l'éléphant

i.

- β : Le lien d'amitié qui existe entre l'éléphant matriarche et l'éléphant i.

III.F.4. La position [17]

Cette étape permet de calculer la position de chaque instance par rapport à chaque classe par l'intermédiaire de l'équation 4.22 :

$$E_{t+1}^i(g) = \frac{E_t^i(g)}{VE_{t+1}^i(g)}$$

- $E_t^i(g)$: position de l'instance i à l'instant T dans la classe g.
- $VE_{t+1}^i(g)$: vitesse de l'instance i à l'instant T+1 dans la classe g.

III.F.5. Évaluation [17]

Chaque instance i est classée dans la classe avec la plus petite position.

III.F.6. Mise à jour [17]

Après chaque itération les paramètres du CEAS sont mises à jour.

III.F.7. Procédure [17]

L'algorithme III.1 résume le fonctionnement du CEAS.

1: Eléphant : instance

2: Entrées :

3: - Mesure de distance

4: - Ensemble de données (base d'apprentissage, base de teste)

5: - Initialisation $(E^i_{T=0}, V^i_{T=0})$

6: $T \leftarrow 0$

7: **while** *not* CD **do**

8: **for** chaque instance a classifier **do**

9: **for** chaque classe g **do**

10: calculer

11: $Mt(g) = \left(\max\left(score(i)\right)\right)g$

12: Trouver l'instance le plus proche du barycentre avec la meilleur position (ME)

13: $VE^i_{\mathrm{T}+1}(g) = \frac{VE^i_{\mathrm{T}}}{\alpha(|ME^g_{\mathrm{T}} - E^i_{\mathrm{T}}|) + \beta(|PE^g_{\mathrm{T}} - E^i_{\mathrm{T}}|)}$

14: $E^i_{t+1}(g) = \frac{E^i_t(g)}{VE^i_{t+1}(g)}$

15: **end for**

16: $L'instance(i) \leftarrow la - classe - avec - la - plus - petite - position$

17: **end for**

18: Mise à jour (ME, PM, V)

19: $T \leftarrow T + 1$

20: **end while**

21: Sortie : Les instances et leur classe.

Algorithme III.1 – Classificateur des éléphants d'Asie sociaux (CEAS)

III.G. Conclusion

La modélisation par biomimétisme a pris sa place dans le domaine informatique pour résoudre des problèmes réels en s'inspirant de la nature et spécialement des organismes vivants. Les éléphants d'Asie forment un modèle de cette inspiration. La vie sociale des éléphants d'Asie est caractérisée par deux phénomènes qui sont les liens d'amitié qui permettent la communication entre les membres du troupeau et la recherche de nourriture et d'eau. De ce dernier phénomène qui est la recherche sont inspirés des algorithmes de recherche d'informations.

Dans ce troisième chapitre, nous avons vu comment les éléphants sociaux d'Asie vivent en groupes, comment ils communiquent entre eux pour atteindre le lieu où se trouve la nourriture et l'eau. Comprendre le mode de vie de ces individus a pour objectif d'inspirer une modélisation pour le processus de recherche d'informations. Dans le chapitre suivant, on procède à concevoir un système de recherche d'informations après avoir comprendre le passage du processus naturel au processus artificiel.

Quatrième chapitre
Expérimentation et résultats

IV. Expérimentation et résultats

IV.A. Introduction

La catégorisation de texte est le processus d'identification de la classe à laquelle appartient un document de texte. Cela implique généralement l'apprentissage, pour chaque classe, de sa représentation à partir d'un ensemble de documents connus pour être membres de cette classe.

Un certain nombre de techniques de classification statistique et d'apprentissage automatique ont été appliquées à la catégorisation des textes, y compris les modèles de régression, *k* plus proches voisins et le modèle naïve bayésienne.

IV.B. Système bio-inspiré pour la catégorisation de texte

À la suite des problèmes de catégorisation de textes, nous avons proposé un algorithme inspiré de comportement des éléphants sociaux d'Asie.

Dans ce chapitre nous traitons le problème de de catégorisation de textes avec l'algorithme des éléphants d'Asie sociaux en utilisant le corpus *20NewsGroup* pour l'expérimentation et les résultats obtenue seront comparés avec un autre classifier.

IV.C. Le corpus 20NewsGroup

L'ensemble de données 20 Newsgroups[5] est une collection d'environ 20 000 documents de groupes de discussion, partitionnés (presque) de manière égale entre 20 groupes de discussion différents. Il a été recueilli à l'origine par Ken Lang, probablement pour son *Newsweeder* : Apprendre à filtrer *netnews* papier, bien qu'il ne mentionne pas explicitement cette collection. La collection de *20newsgroups* est devenue un ensemble de données populaire pour les expériences dans les applications de texte de techniques d'apprentissage automatique, telles que la classification de texte et le regroupement de texte.

IV.C.1. Organisation du corpus 20NewsGroup

Les données sont organisées en 20 groupes de discussion différents, correspondant chacun à un sujet différent. Certains des groupes de discussion sont très proches les uns des autres (par exemple *comp.sys.ibm.pc.hardware/comp.sys.mac.hardware*), tandis que d'autres sont hautement indépendants (par exemple *misc.forsale/soc.religion.christian*). Voici une liste des 20newsgroups, partitionnée (plus ou moins) en fonction du sujet :

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

[5] Le corpus 20NewsGroup est disponible dans : http://qwone.com/~jason/20Newsgroups/

Tableau IV.1 – Organisation du corpus 20NewsGroup

IV.D. Prétraitement des données textuelles

IV.D.1. Nettoyage

Les mots vides apparaissent très souvent dans tous les textes comme les articles, les pronoms, les prépositions, les complimenteurs et les déterminants. Ils constituent une grande part des mots d'un texte, mais ils sont faiblement informatifs. L'élimination de ces mots est effectuée par l'intermédiaire d'une liste prédéfinie de mots vides pour chaque langue étudiée comme l'expose l'exemple suivant. À titre d'exemple nous pouvons citer pour le français[6] les mots {le, une, les, des, ainsi, ensuite, toutefois, des, . . . etc.} et pour l'anglais[7] {the, that, after, one, are, above, few, … etc}.

IV.D.2. Représentation de texte

Cette étape transforme les textes vers une liste de termes. Dans notre travail nous avons utilisé différentes techniques de représentation comme :

IV.D.2.a. Représentation par sac de mots

Le sac de mots découpe les séquences de caractères liés en fonction de la présence ou l'absence des caractères de séparation de type espace, tabulation ou retour à la ligne. Elle transforme les textes vers une liste de mots appelée sac, par exemple la phrase « Je suis étudiant à l'université » deviendra une liste des mots {je, suis, étudiant, à, université}.

IV.D.2.b. Représentation par racinisation (stemming)

Cette technique est basée sur le regroupement des mots ayant la même racine (stem). Pour l'extraction des racines nous avons appliqué l'algorithme de porter basé sur des règles de remplacement des chaînes de caractères pour supprimer les suffixes les plus utilisées et les signes du pluriel, par exemple le mot « troubling » deviendra « trouble » ou « relation» deviendra « relate » [29].

IV.D.3. Codage

Cette étape calcule la fréquence (importance) de chaque attribut (composant) dans chaque document en utilisant une pondération liée au texte lui-même (ex : le nombre d'occurrences d'un terme dans le texte) et à l'ensemble de données en intégralité (ex : le nombre d'occurrences du terme dans l'ensemble des données).

Pour notre travail nous avons utilisé la pondération TF*IDF :

IV.D.3.a. La pondération TF*IDF

La fonction Tf-IDF (acronyme pour « term frequency inverse document frequency ») est la pondération la plus utilisée dans la littérature. Sa force réside dans le fait qu'elle implémente en même temps : l'Exhaustivité et Spécificité.

Le poids de terme *tk* appartenant au document *di* égale à :

$$Tf - IDF(t_k, d_i) = N * \log(\frac{A}{B})$$

[6] La liste des mots vide français est disponible dans : http ://www.ranks.nl/stopwords/french

[7] La liste des mots vide anglais est disponible dans : http ://www.ranks.nl/stopwords

Où :

- N : le nombre d'occurrences du terme *tk* dans le texte *d* ;
- A : le nombre total de textes du corpus ;
- B : le nombre de textes dans lesquels le terme *tk* apparaît au moins une fois.

Si on désire avoir des poids entre 0 et 1, on peut la normaliser. La fonction T*f* ¡IDF a deux points forts : efficacité dans des tâches de catégorisation de textes et simplicité de calcul. Cette pondération issue du domaine de Recherche d'Information est inspirée de la loi de Zipf.

IV.E. Évaluation

Nous avons utilisé un ensemble de mesures d'évaluation pour la validation de notre travail

Les mesures d'évaluation sont basées sur une matrice de confusion telle que présentée dans le *Tableau IV.2*, qui fournit quatre informations essentielles pour la comparaison entre la classification de notre algorithme et la classification d'un expert (pré-classification).

Matrice de contingence		Jugement de l'expert	
		Vrais	Faux
Jugement de notre système	Vrais	VP_i	FP_i
	Faux	FN_i	VN_i
Vrais positive (VP):	Le nombre d'instances attribués à une catégorie convenablement. (instances attribués à leurs vraies catégories)		
Vrais Négative (VN):	Le nombre d'instances non attribués à une catégorie convenablement (Qui n'ont pas à être attribués à une catégorie, et ne l'ont pas été)		
Faux positive (FP):	Le nombre d'instances attribués à une catégorie inconvenablement. (instances attribués à des mauvaises catégories)		
Faux négative (FN):	Le nombre d'instances inconvenablement non attribués. (Qui auraient dû être attribués à une catégorie mais qui ne l'ont pas été)		

Tableau IV.2 – Matrice de confusion

IV.E.1. Rappel (R)

Le rappel mesure la capacité de notre système à détecter les instances bien classées. Comme la montre l'équation (1), le R représente le rapport entre le nombre de documents correctement classés par notre système dans la classe Ci par rapport au nombre total des documents réellement dans la classe c.

$$R = \frac{\mathrm{VP_i}}{\mathrm{VP_i} * FN_i} (1)$$

IV.E.2. Précision (P)

La précision permet de mesurer la capacité d'un système à retourner seulement les instances bien classées. Comme la montre l'équation (2) la P représente le rapport entre le

nombre d'instances correctement classé par l'algorithme dans la classe Ci par rapport au nombre d'instances classées par notre système dans la classe Ci.

$$P = \frac{\mathrm{VP_i}}{\mathrm{VP_i} * FP_i} (2)$$

IV.E.3. F-mesure (F)

Comme la montre l'équation (3) la f-mesure permet de calculer la qualité de classification d'un algorithme à partir du rappel et de la précision.

$$F = \frac{2 * \mathrm{R} * \mathrm{P}}{(\mathrm{R}+\mathrm{P})} (3)$$

IV.E.4. Taux de succès (TS)

Comme le montre l'équation (4). Le TS appelé accuracy en anglais permet de calculer l'exactitude d'un algorithme. Elle représente le pourcentage des instances correctement classé.

$$TS = \frac{\mathrm{VP_i}+\mathrm{VN_i}}{\mathrm{VP_i}+\mathrm{VN_i}\mathrm{FP_i}+\mathrm{FN_i}} (4)$$

IV.F. Notre approche

L'objectif de notre travail consiste à implémenter un système de recherche d'informations basé sur la modélisation par les éléphants d'Asie. Puis le comparer avec d'autres algorithmes de datamining. Dans le cadre de notre expérimentation, cette comparaison porte à utiliser le logiciel Weka, choisir quelques algorithmes et comparer les résultats obtenus avec ceux obtenus par notre système de recherche d'information.

La base de travail initiale du projet consiste en une base de 500 documents textuels issus de la base de données *20NewsGroups*, de taille et de nature diverses (e-mail, article, dépêche, ...etc).

L'extraction des 500 documents et la construction de la base d'apprentissage a été faite comme suit :

- Classe A = alt.atheism = 50 Document s
- Classe B = comp.graphics = 50 Document s
- Classe C = comp.windows.x = 50 Document s
- Classe D = rec.motorcycles = 50 Document s
- Classe E = rec.sport.baseball = 50 Document s
- Classe F = rec.sport.hockey = 50 Document s
- Classe G = sci.crypt = 50 Document s
- Classe H = sci.electronics = 50 Document s
- Classe I = sci.space = 50 Document s
- Classe G = talk.politics.guns = 50 Document s

IV.G. Matériel et logiciels utilisés

Pour procéder à cette expérimentation, nous avons utilisé la plateforme matériel et logiciel que nous allons détailler ci-dessous :

IV.G.1.a. Hardware

Un PC de marque DELL dont les caractéristiques sont :

- Processeur : Intel® Core ™ i5(4200U) CPU @1.60 GHz 2.30 GHz ;
- Mémoire : 6.00 Go ;
- Disque Dur : 1 To.

IV.G.1.b. Software

- Un système d'exploitation Windows 7 Professionnel Service Pack 1 ;
- Environnement de développement Java 8 ;
- Environnement de Développement Intégré (IDE Eclipse Néon) ;
- Une Interface Graphique d'Utilisateur GUI JavaFX ;
- Weka version 3.8.1 (pour comparer les résultats).

IV.G.1.c. Présentation de l'application

Lors de l'exécution de notre application, une fenêtre de bienvenue va-t-être affichée comme suit :

Figure IV.1 – Interface de bienvenue

Cette interface dune quelques secondes, puis apparaît l'interface principale de l'application :

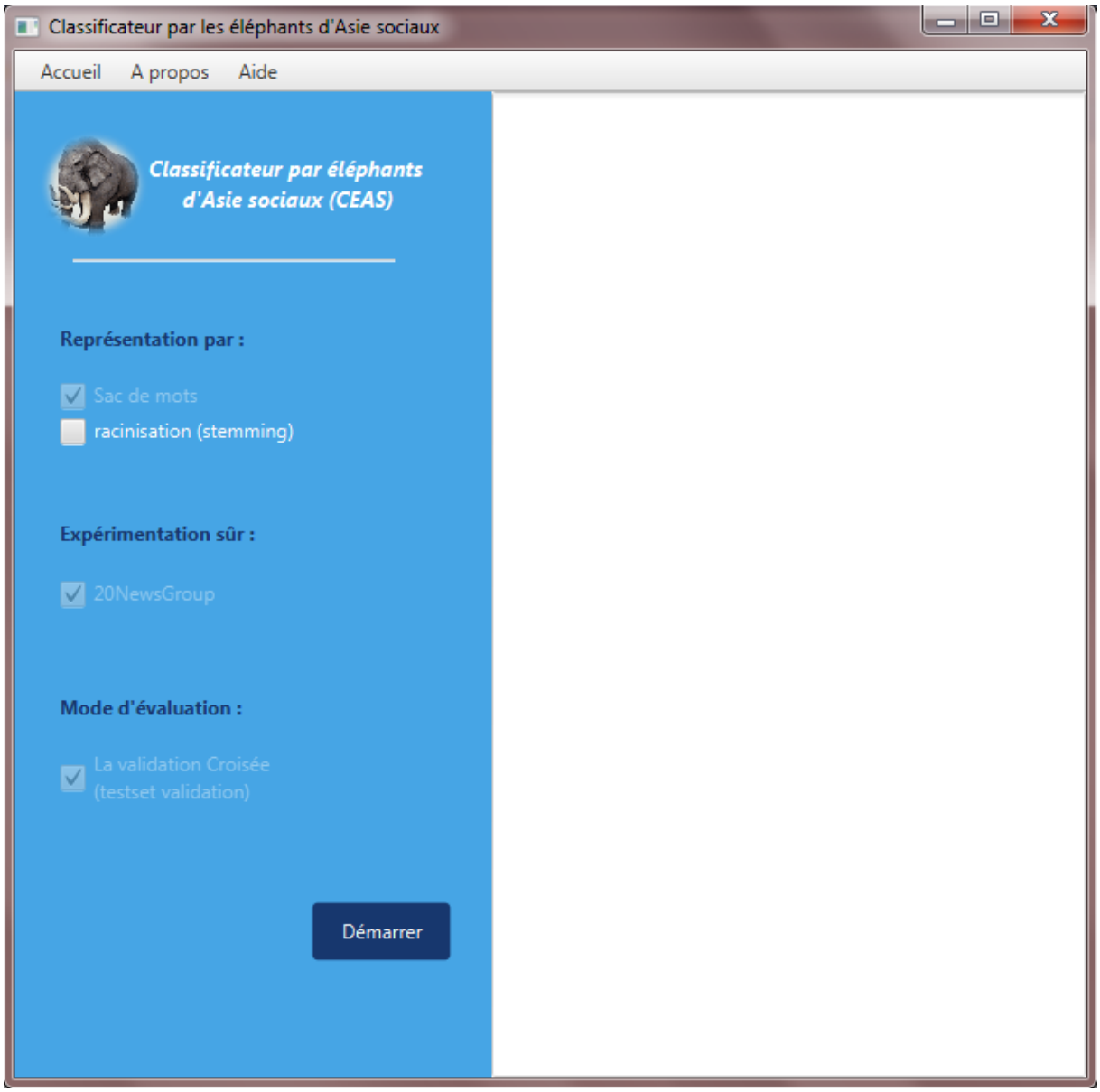

Figure IV.2 – Interface principale de l'application

Par défaut, la méthode représentation est : sac de mots, et le corpus choisi est *20Newsgroup* comme mentionné auparavant, en cliquant sur le bouton *Démarrer*, les résultats obtenus sont comme suit :

IV.H. Résultats de l'expérimentation

Résultat obtenus en utilisant la représentation par sac de mots :

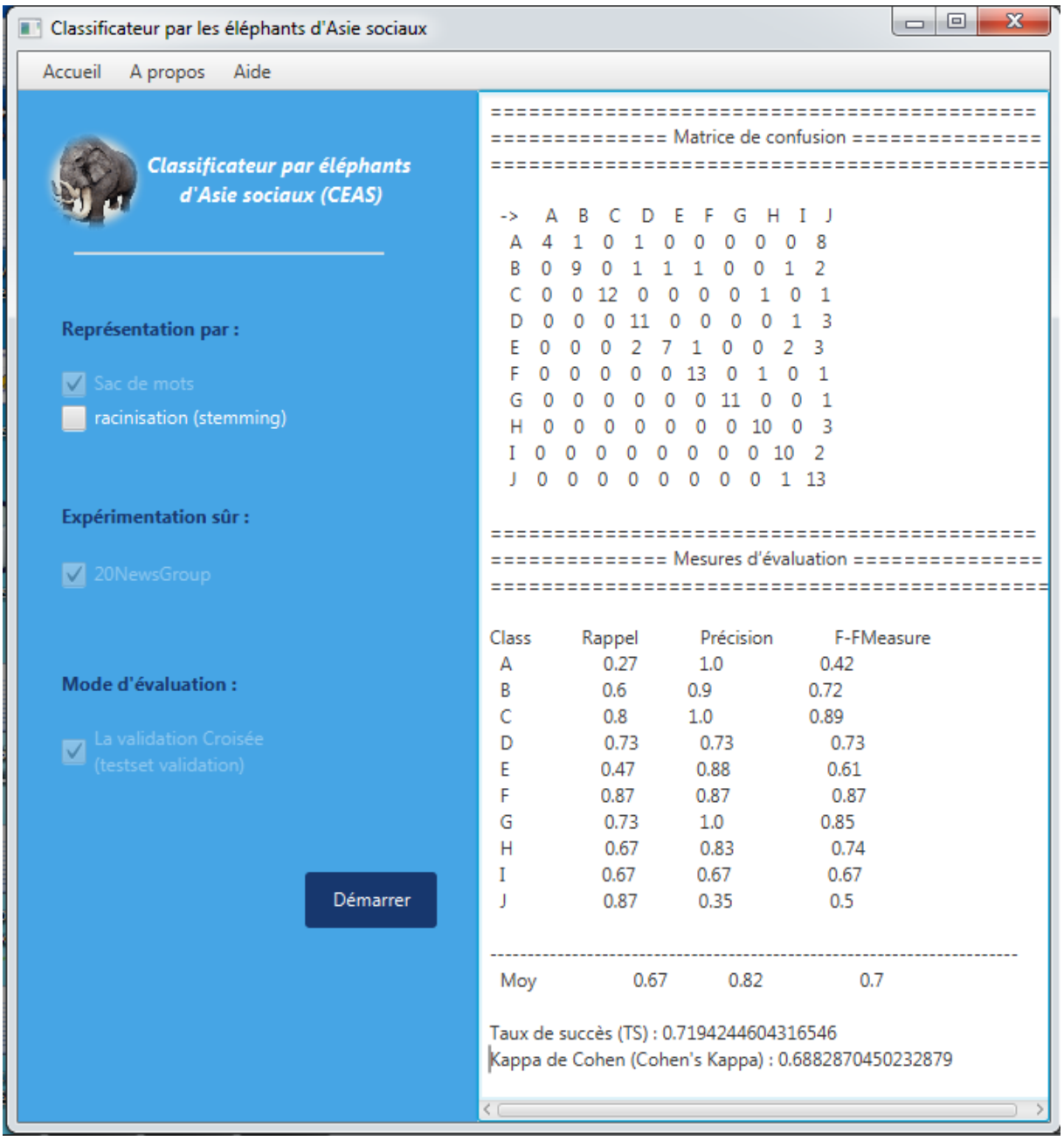

Figure IV.3 – Résultats obtenus en utilisation la représentation sac de mots

Résultat obtenus en utilisant la racinisation :

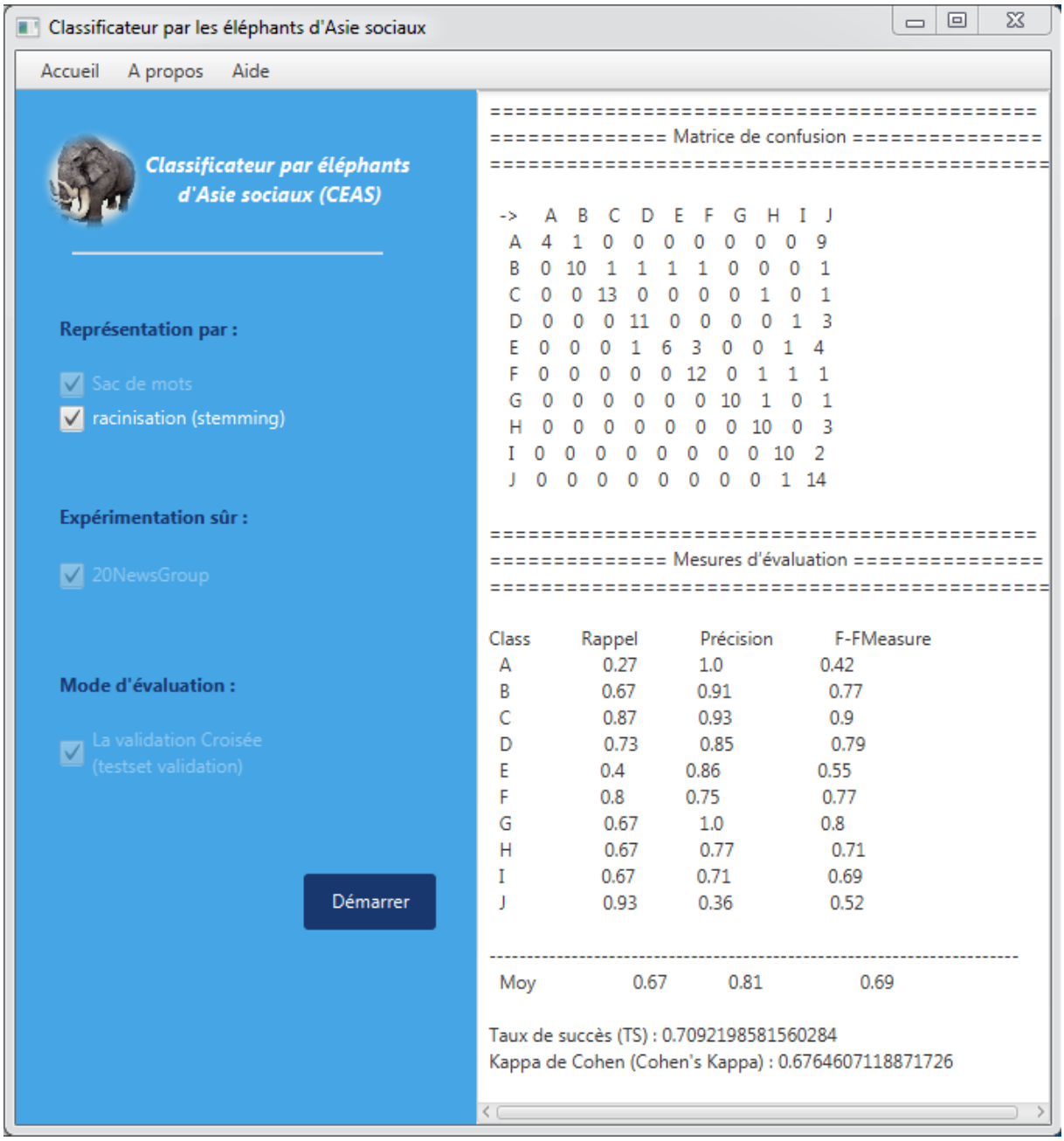

Figure IV.4 - Résultats obtenus en utilisant la racinisation

IV.I. Expérimentation par Weka

IV.I.1. Weka

Weka (acronyme pour *Waikato Environment for knowledge Analysis*, en français : « environnement Waikato pour l'analyse de connaissances ») est une suite de logiciels d'apprentissage automatique. Écrite en Java, développée à l'université de Waikato en Nouvelle-Zélande. Weka est un logiciel libre disponible sous la Licence publique générale GNU (GPL).

Figure IV.5 – Interface de Weka version 3.8.1

IV.I.2. L'acquisition des données

Les fichiers ARFF sont le format principal pour utiliser n'importe quelle tâche de classification dans WEKA. Ces fichiers ont considéré les données d'entrée de base (concepts, instances et attributs) pour l'exploration de données. Donc on est besoins de convertir notre collection des documents à un fichier ARFF.

Pour expérimente notre travail avec Weka on à convertir notre corpus en format ARFF.

IV.I.3. Résultats obtenus en utilisant NaiveBayes

```
=== Summary ===

Correctly Classified Instances         116               77.3333 %
Incorrectly Classified Instances        34               22.6667 %
Kappa statistic                          0.7483
Mean absolute error                      0.0451
Root mean squared error                  0.208
Relative absolute error                 25.0498 %
Root relative squared error             69.2508 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,917    0,022    0,786      0,917    0,846      0,835    0,986     0,875     alt.atheism
                 0,714    0,125    0,370      0,714    0,488      0,446    0,900     0,605     comp.graphics
                 0,533    0,000    1,000      0,533    0,696      0,712    0,970     0,840     comp.windows.x
                 0,857    0,000    1,000      0,857    0,923      0,919    0,994     0,957     rec.motorcycles
                 0,929    0,007    0,929      0,929    0,929      0,921    0,986     0,953     rec.sport.baseball
                 0,778    0,000    1,000      0,778    0,875      0,869    0,997     0,979     rec.sport.hockey
                 0,842    0,023    0,842      0,842    0,842      0,819    0,970     0,919     sci.crypt
                 0,833    0,072    0,500      0,833    0,625      0,607    0,971     0,842     sci.electronics
                 0,667    0,000    1,000      0,667    0,800      0,802    0,985     0,908     sci.space
                 0,706    0,000    1,000      0,706    0,828      0,825    0,996     0,981     talk.politics.guns
Weighted Avg.    0,773    0,023    0,857      0,773    0,791      0,782    0,976     0,892

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
 11  1  0  0  0  0  0  0  0  0 |  a = alt.atheism
  1 10  0  0  0  0  0  3  0  0 |  b = comp.graphics
  0  4  8  0  0  0  0  3  0  0 |  c = comp.windows.x
  1  1  0 12  0  0  0  0  0  0 |  d = rec.motorcycles
  0  0  0  0 13  0  1  0  0  0 |  e = rec.sport.baseball
  0  3  0  0  1 14  0  0  0  0 |  f = rec.sport.hockey
  0  2  0  0  0  0 16  1  0  0 |  g = sci.crypt
  0  2  0  0  0  0  0 10  0  0 |  h = sci.electronics
  0  2  0  0  0  0  0  3 10  0 |  i = sci.space
  1  2  0  0  0  0  2  0  0 12 |  j = talk.politics.guns
```

Figure IV.6 – Résultats obtenus par NaiveBayes

IV.I.4. Résultats obtenus par J48

```
=== Summary ===

Correctly Classified Instances         147               98      %
Incorrectly Classified Instances         3                2      %
Kappa statistic                          0.9777
Mean absolute error                      0.004
Root mean squared error                  0.0632
Relative absolute error                  2.2199 %
Root relative squared error             21.0557 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     alt.atheism
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     comp.graphics
                 1,000    0,007    0,938      1,000    0,968      0,965    0,996     0,938     comp.windows.x
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     rec.motorcycles
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     rec.sport.baseball
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     rec.sport.hockey
                 0,947    0,000    1,000      0,947    0,973      0,970    0,974     0,954     sci.crypt
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     sci.electronics
                 1,000    0,015    0,882      1,000    0,938      0,932    0,993     0,882     sci.space
                 0,882    0,000    1,000      0,882    0,938      0,932    0,941     0,896     talk.politics.guns
Weighted Avg.    0,980    0,002    0,982      0,980    0,980      0,978    0,989     0,964

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
 12  0  0  0  0  0  0  0  0  0 |  a = alt.atheism
  0 14  0  0  0  0  0  0  0  0 |  b = comp.graphics
  0  0 15  0  0  0  0  0  0  0 |  c = comp.windows.x
  0  0  0 14  0  0  0  0  0  0 |  d = rec.motorcycles
  0  0  0  0 14  0  0  0  0  0 |  e = rec.sport.baseball
  0  0  0  0  0 18  0  0  0  0 |  f = rec.sport.hockey
  0  0  1  0  0  0 18  0  0  0 |  g = sci.crypt
  0  0  0  0  0  0  0 12  0  0 |  h = sci.electronics
  0  0  0  0  0  0  0  0 15  0 |  i = sci.space
  0  0  0  0  0  0  0  0  2 15 |  j = talk.politics.guns
```

Figure IV.7 – Résultats obtenus par J48

IV.I.5. Résultats obtenus par KNN

```
=== Summary ===

Correctly Classified Instances          70               46.6667 %
Incorrectly Classified Instances        80               53.3333 %
Kappa statistic                          0.4099
Mean absolute error                      0.1395
Root mean squared error                  0.2551
Relative absolute error                 77.441  %
Root relative squared error             84.9293 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,583    0,007    0,875      0,583    0,700      0,696    0,835     0,703     alt.atheism
                 0,929    0,265    0,265      0,929    0,413      0,412    0,939     0,709     comp.graphics
                 0,667    0,015    0,833      0,667    0,741      0,721    0,959     0,862     comp.windows.x
                 0,286    0,000    1,000      0,286    0,444      0,516    0,911     0,678     rec.motorcycles
                 0,929    0,301    0,241      0,929    0,382      0,380    0,957     0,815     rec.sport.baseball
                 0,056    0,000    1,000      0,056    0,105      0,222    0,926     0,718     rec.sport.hockey
                 0,421    0,000    1,000      0,421    0,593      0,623    0,826     0,566     sci.crypt
                 0,083    0,000    1,000      0,083    0,154      0,278    0,944     0,578     sci.electronics
                 0,400    0,000    1,000      0,400    0,571      0,612    0,881     0,672     sci.space
                 0,412    0,000    1,000      0,412    0,583      0,619    0,916     0,736     talk.politics.guns
Weighted Avg.    0,467    0,055    0,834      0,467    0,469      0,509    0,908     0,703

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
  7  3  0  0  2  0  0  0  0  0 |  a = alt.atheism
  0 13  0  0  1  0  0  0  0  0 |  b = comp.graphics
  0  5 10  0  0  0  0  0  0  0 |  c = comp.windows.x
  0  2  0  4  8  0  0  0  0  0 |  d = rec.motorcycles
  0  1  0  0 13  0  0  0  0  0 |  e = rec.sport.baseball
  0  3  0  0 14  1  0  0  0  0 |  f = rec.sport.hockey
  0  6  0  0  5  0  8  0  0  0 |  g = sci.crypt
  0  6  0  0  5  0  0  1  0  0 |  h = sci.electronics
  0  4  1  0  4  0  0  0  6  0 |  i = sci.space
  1  6  1  0  2  0  0  0  0  7 |  j = talk.politics.guns
```

Figure IV.8 – Résultats obtenus par KNN

IV.J. Comparaison et discussion des résultats

Après une étude comparative entre notre classifieur et un ensemble d'autres classifieurs statiques en utilisant Weka, les résultats obtenus montrent une remarquable réussite de notre classifieur CEAS qui est inspiré de la vie biologique des Eléphants d'Asie Sociaux avec un taux de succès de 72% des instances correctement classées comme montre la *Figure IV.3*.

Comme perspective, on veut implémenter d'autre technique de représentation des textes tel que la représentation n-gram caractères et n-gram mots et la représentation en sac de phrases, on veut aussi d'ajouter des méthodes de réduction de dimensionnalité pour éliminer les termes non informatifs afin d'obtenir des meilleurs résultats et de réduire le temps de réponse par notre système.

Le tableau suivant montre les résultats obtenus par les classificateurs : Naïve Bayes, KNN (K-plus proches voisins), et par arbre de décision (J48) :

Algorithme	Rappel	Précision	F-Measure
Classificateur par les éléphants d'Asie sociaux	0.67	0.82	0.7
Naive bayes	0.773	0.857	0.791
KNN	0.467	0.834	0.469
J48	0.980	0.982	0.980

Tableau IV.3 – Résultats de comparaison entre le classificateurs des éléphants d'Asie sociaux et d'autres classificateurs

D’après les résultats que montre le *Tableau IV.3*, le meilleur classificateur est le classificateur par arbre de décision qui a donné une précision de 0.982 par rapport à notre classificateur les éléphants sociaux d’Asie et même par rapports aux autres algorithmes (KNN et Naïve Bayes). Cela indique que les documents bien classés par le classificateur J48 avaient un taux de 0.98 %.

V. Conclusion générale

Observer la nature pour l'imiter est une idée ancienne. L'être vivant a développé des stratégies adaptatives et durables dont s'inspirent les sciences et les technologies. Plusieurs recherches et applications nées de cette démarche en pleine expansion : le biomimétisme.

Comment faire face à des problèmes complexes en inspirant de cette nature et comment traduire le processus naturel en un processus artificiel ? Plusieurs travaux basés sur cette inspiration ont été réalisés dans des domaines différents. La recherche d'information en fait partie.

Avec l'apparition du web et de cette masse de données, trouver une information pertinente dans un temps bref est l'objectif des recherches actuelles. Plusieurs systèmes de RI existent actuellement. Concevoir un système de recherche d'information qui répond aux exigences de l'utilisateur et assure la performance et la pertinence est l'une des défis de la recherche en informatique.

Notre étude a été portée sur la modélisation du processus de recherche d'information par biomimétisme en inspirants des éléphants sociaux d'Asie. Ces individus qui vivent en troupeau et qui se chartérisent par deux phénomènes qui sont les liens d'amitié et la recherche d'eau d'où on inspire sa manière de recherche.

Notre expérimentation basée sur le système par éléphants sociaux d'Asie a abouti à des résultats inferieur aux résultats obtenus par l'applications d'autres algorithmes implémentés par le logiciel Weka, les arbres de décision (J48) ont prouvé leur efficacité par rapport à notre système implémenté.

Bibliographie

[1] Abdelkrim BOURAMOUL. «Recherche d'information contextuelle et sémantique.» Thèse de doct., Université de Constantine, 2011.

[2] Ali LEMOUARI. «Introduction aux Métaheuristiques.» support de cours, Ecole Polytechnique de Montréal, France, 2014.

[3] Amine BOUMAZA et Bruno SCHERRER. «Analyse d'un algorithme d'intelligence en essaim pour le fourragement.» *Revue des Sciences et Technologies de l'Information-Série RIA: Revue d'Intelligence Artificielle 22.6*, 2008: 791–816.

[4] Calvin N MOOERS. *Information retrieval viewed as temporal signaling.* Vol. 1. 1950: Proceedings of the international congress of mathematicians, s.d.

[5] Carlos M FONSECA, Peter J FLEMING et al. «Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization.» *ICGA. T. 93. Citeseer.*, 1993: 416-423.

[6] Catherine LELOU. Moteurs d'indexation et de recherche : environnement client-serveur, Internet et Intranet. Paris: Eyrolles, 1998.

[7] César REGO et Bahram ALIDAEE. «Metaheuristic optimization via memory and evolution: tabu search and scatter search.» *Springer Science & Business Media* 30 (2006).

[8] Christos H PAPADIMITRIOU et Kenneth STEIGLITZ. «Combinatorial optimization: algorithms and complexity.» *Courier Corporation*, 1982.

[9] David Edward GOLDBERG et al. Algorithmes génétiques: exploration, optimisation et apprentissage automatique. Addison-Wesley France, 1994.

[10] Djalila BOUGHAREB. «Recherche d'information multicritères.» Thèse de doct., Université Badji Mokhtar Annaba, 2014.

[11] Eric BONABEAU, Marco DORIGO et Guy THERAULAZ. «Swarm intelligence: from natural to artificial systems.» *Oxford university press*, 1999.

[12] F. BOUBEKEUR. «Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets.» thèse de doctorat en informatique, Université Paul Sabatier, 2008.

[13] Fred GLOVER et Manuel LAGUNA. «Tabu search.» *Kluwer Academic Publishers*, 1997.

[14] G. Salton and M. McGill. «Introduction to Modern Information Retrieval.» *New York*, 1983.

[15] Gérard SALTON. «A comparison between manual and automatic indexing methods.» *Journal of American Documentation*, 1971: 61-71.

[16] Gerardo BENI, Jing WANG. «Swarm intelligence in cellular robotic systems.» *Robots and Biological Systems: Towards a New Bionics? Springer*, 1993: 703–712.

[17] Hadj Ahmed BOUARARA. «Méta-heuristique et les techniques bio-inspirées dans la recherche.» Thèse de doct., Université Dr Tahar Moulay, Saïda, 2017.

[18] Johann DREO et al. «Metaheuristics for hard optimization: methods and case studies.» *Springer Science & Business Media*, 2006.

[19] Jordi DELGADO et Ricard V SOLÉ. «Collective-induced computation.» *Physical Review*, 1997: 2338.

[20] Joyce H POOLE. «Signals and assessment in African elephants : evidence from playback experiments.» *Animal Behaviour 58.1*, 1999: 185–193.

[21] Karen MCCOMB et al. «Matriarchs as repositories of social knowledge in African elephants.» *Science 292.5516*, 2001: 491–494.

[22] Lucy A BATES et al. «Elephants classify human ethnic groups by odor and garment color.» *Current Biology 17.22*, 2007: 1938–1942.

[23] M. E. Maron and J. L. Kuhns. «On relevance, probabilistic indexing and information retrieval.» *Journal. ACM, 7(3)*, 1960: 216–244.

[24] Mélina MAIORANO. «Approche biomimetique dans la conception d'un reseau de transport : application a un systeme de transport cybernetique.» Montreal, octobre 2013.

[25] Michael R GARY et David S JOHNSON. Computers and Intractability: A Guide to the Theory of NP-completeness. 1979.

[26] Mitchell KAPOR et Daniel J WEITZNER. «Developing the national communications and information infrastructure.» *Internet Research 20.4*, 2010: 395–407.

[27] Nathalie HERNANDEZ. «Ontologie de domaine pour la modélisation du contexte en recherche d'information.» thèse de doctorat en informatique, Université Paul Sabatier, 2006.

[28] Norbert FUHR. «Information Retrieval-From Information Access to Contextual Retrieval.» *Designing Information Systems*, 2004: 47–57.

[29] Peter WILLETT. «The Porter stemming algorithm : then and now.» *Program 40.3*, 2006: 219–223.

[30] Rainer STORN et Kenneth PRICE. «Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.» *Journal of global optimization*, 1997: 341–359.

[31] Ricardo BAEZA-YATES, Berthier RIBEIRO-NETO et al. «Modern information retrieval.» *ACM press New York* 463 (1999).

[32] Richard William BYRNE, Lucy BATES et Cynthia J MOSS. «Elephant cognition in primate perspective.» *Comparative Cognition & Behavior Reviews 4.2*, 2009: 65– 79.

[33] Rokia BENDAOUD. «Analyses formelle et relationnelle de concepts pour la construction d'ontologies de domaines à partir de ressources textuelles hétérogènes.» Thèse de doct., Université Henri Poincaré - Nancy, Juillet 2009, 15.

[34] S. Robertson and K. Sparck Jones. «Relevance weighting for search terms.» *Journal of The American Society for Information Science*, 1976: 129–146.

[35] Scott KIRKPATRICK. «Optimization by simulated annealing: Quantitative studies.» *Journal of statistical physics*, 1984: 975–986.

[36] Soheila KARBASI. «Pondération des termes en Recherche d'Information.» Thèse de doct., Université Paul Sabatier, 2007.

[37] Thomas BÄCK, David B FOGEL et Zbigniew MICHALEWICZ. «Evolutionary computation 1: basic algorithms and operators.» *CRC Press"* T.1 (2000).

[38] Thomas BÄCK, DB FOGEL et Z MICHALEWICZ. «Handbook of evolutionary computation.» *Release 97.1*, 1997.