Un Algorithme génétique spécifique à une reformulation multi-requêtes dans un système de recherche d'information

L. Tamine, M. Boughanem

IRIT-SIG,
Université Paul Sabatier
118, Route de Narbonne
31062 Toulouse Cedex
lechani@wissal.dz, bougha@irit.fr

RESUME . Cet article présente une approche de reformulation de requête fondée sur l'utilisation combinée de la stratégie d'injection de pertinence et des techniques avancées de l'algorithmique génétique. Nous proposons un processus génétique d'optimisation multi-requêtes amélioré par l'intégration des heuristiques de nichage et adaptation des opérateurs génétiques. L'heuristique de nichage assure une recherche d'information coopérative dans différentes directions de l'espace documentaire. L'intégration de la connaissance à la structure des opérateurs permet d'améliorer les conditions de convergence de l'algorithme. Nous montrons, à l'aide d'expérimentations réalisées sur une collection TREC, l'intérêt de notre approche.

ABSTRACT. The paper presents a query reformulation approach using both relevance feedback and advanced techniques of genetic algorithms. We propose a genetic process improved by the use of niching technique and knowledge based operators. The niching technique allows a cooperative retrieval processing in different directions of the document space. The integration of domain knowledge in the structure of the genetic operators improve the convergence conditions of the algorithm. We show the effectiveness of our approach, using experiments done with a TREC collection.

MOTS-CLES : Système de recherche d'information, reformulation de requête, algorithme génétique
KEY WORDS : Information retrieval system, query reformulation, genetic algorithm

1. Introduction

L'intérêt stratégique porté à l'information, combiné à l'avènement explosif d'Internet et autres services de l'information, sont des facteurs déterminants qui justifient la multiplication de directions de recherche ayant pour objectif de mettre en œuvre des processus automatiques d'accès à l'information, sans cesse plus performants.

Dans un contexte très large, un système de recherche d'information capitalise un volume considérable d'informations et offre des outils et techniques permettant de localiser l'information pertinente relativement au besoin en information exprimé par l'utilisateur. Dans un contexte précis, la recherche documentaire est une activité quotidienne très largement pratiquée par diverses catégories d'utilisateurs. Un Système de Recherche d'Information (SRI) manipule dans ce cas, une collection de documents traduisant des connaissances hétérogènes et indépendantes qu'il convient d'homogénéiser à travers la découverte d'associations sémantiques, dans le but de structurer la réponse au besoin exprimé par l'utilisateur.

Le processus d'appariement d'une requête et d'un document pose les principaux problèmes suivants :

- le besoin en information formulé par une requête utilisateur est généralement imprécis, il s'ensuit que l'objet de la recherche d'information est à priori inconnu,
- les univers de référence des auteurs et utilisateurs sont différents,
- les procédures d'indexation automatique ne résolvent pas totalement les propriétés d'ambiguïté et de recouvrement de concepts connus en langage naturel,
- la notion de pertinence dépend étroitement de l'utilisateur.

La conséquence immédiate à cet ensemble de difficultés est qu'un ensemble de documents pertinents à la requête utilisateur n'est pas sélectionné par le SRI.
Dans ce cadre, de nombreux modèles de recherche et de représentation d'information ont été proposés : vectoriel [35], [36], probabiliste [31], LSI [11], connexionniste [24], [43].
Outre ces modèles formels, de nombreux travaux ont investi la mise en œuvre de stratégies qui représentent un ensemble d'heuristiques et algorithmes, greffés à un processus de recherche de base, afin d'en améliorer les performances.
La stratégie la plus largement adoptée est la reformulation de requête [34][2] [16] [33] [6].
C'est un processus permettant de générer une requête plus adéquate à la recherche d'information dans l'environnement du SRI, que celle initialement formulée par l'utilisateur. Son principe fondamental est de modifier la requête de l'utilisateur par l'ajout de termes significatifs, par la ré-estimation de leur poids d'indexation ou par la combinaison de ces deux techniques.
La dimension de l'espace de recherche étant élevée, la principale difficulté de la reformulation de requête est alors la définition de l'approche à adopter en vue de la réduire.
Dans ce contexte on recense particulièrement l'utilisation combinée de thésaurus et de techniques de classification [38] [30] [37] [20], résultats de recherche [44] [27] et jugement de pertinence de l'utilisateur [17] [32] [9].
De nombreux travaux utilisent des techniques d'algorithmique génétique pour aborder le problème de la recherche d'information de manière générale, et plus particulièrement la reformulation de requête. C'est le cadre précis de nos travaux.
Les algorithmes génétiques sont des métaphores biologiques inspirées des mécanismes de l'évolution darwinienne et de la génétique moderne, utilisés comme un outil puissant d'optimisation.
Nous exploitons les concepts et techniques de l'algorithmique génétique afin de mettre en œuvre un processus d'optimisation – reformulation de requête caractérisée par une exploration efficace du fond documentaire et une recherche graduelle et coopérative d'information. Notre démarche se caractérise par :

1- l'utilisation de la technique de nichage [13] en vue de restituer des documents pertinents à une même requête mais qui ont des descripteurs relativement dissemblants,
2- l'intégration de techniques de reformulation de requête dans la structure des opérateurs génétiques.

Notons à cet effet que notre objectif essentiel n'est pas tant d'introduire des heuristiques avancées de l'algorithmique génétique ni d'établir un état de l'art sur les travaux dans ce domaine, le lecteur intéressé peut se référer à [13] [22], [12]. Notre but est d'évaluer l'impact de l'adaptation d'une classe d'algorithmes génétiques au contexte de la reformulation de requête dans un SRI.
Dans la suite de cet article, nous présentons tout d'abord un aperçu des techniques de reformulation de requête d'une part et des algorithmes génétiques, en rapport avec notre problématique, d'autre part. Puis nous détaillons la description de notre approche de reformulation de requête. Nous présentons finalement des résultats issus d'expérimentations réalisées à l'aide du SRI Mercure [6] sur la collection AP88 issue de TREC, qui valident notre approche.

2. Reformulation de requête

La reformulation de requête est proposée comme une méthode élaborée pour la recherche d'information, s'inscrivant dans la voie de conception de SRI adaptatifs aux besoins des utilisateurs ; elle met en œuvre un algorithme de modification de la requête en termes, en poids ou les deux simultanément, moyennant des critères de choix de termes d'expansion et règle de calcul de nouveaux poids.
Nous distinguons principalement deux types de reformulation de requête : la reformulation directe et la reformulation par injection de pertinence.

2.1. Reformulation directe

La reformulation directe de requête induit un processus d'expansion et/ou repondération de la requête initiale en utilisant des critères de choix définis sans intervention de l'utilisateur. Ce type de reformulation peut être défini dans un contexte global, basé sur le thésaurus, ou alors local, basé sur les résultats de recherche en cours.
La reformulation basée sur le contexte global est essentiellement basée sur l'ajout et repondération de termes issus d'un thésaurus manuel [38] ou construit automatiquement en utilisant des calculs de poids de similarité [30], cooccurrence [37] et relations contextuelles entre termes [20] ou une combinaison de divers types de relations [25].
La reformulation basée sur le contexte local [44] [27], utilise des informations issues de la recherche en cours : documents retrouvés, termes et poids associés.
Les deux types de reformulation diffèrent essentiellement par la source d'information utilisée quant à la dérivation de l'association sémantique entre termes ou entre termes et documents, mais utilisent toutes les deux des fonctions caractéristiques pour la sélection et repondération des nouveaux termes de la requête.

2.2. Reformulation par injection de pertinence

La reformulation par injection de pertinence est une forme de recherche évolutive et interactive ; elle procède à la modification de la requête initiale en termes et poids, sur la base des jugements de pertinence de l'utilisateur sur les documents restitués par le SRI [34] [9] [17] [32].
Son principe fondamental est d'utiliser la requête initiale pour amorcer la recherche d'information, puis exploiter itérativement les jugements de pertinence de l'utilisateur afin d'ajuster la requête par expansion repondération, ou combinaison des deux procédures, en direction des documents pertinents. Une étude synthétique des travaux afférents [39] nous a permis de dégager trois principaux paramètres de performances : le nombre de termes ajoutés à la requête, la méthode de sélection des termes et la longueur moyenne de la requête initiale.

3. Les algorithmes génétiques

Les algorithmes évolutifs sont des algorithmes stochastiques fondés sur la manipulation du processus d'évolution et d'adaptation des organismes dans les milieux naturels. Dans cette large classe d'algorithmes, on retrouve la sous-classe des algorithmes génétiques [18] [13]. Ces derniers sont des processus d'optimisation de problèmes, fondés sur la théorie darwinienne. Un Algorithme Génétique (AG) a pour but de faire évoluer un ensemble de solutions candidates à un problème posé vers la solution optimale. Cette évolution s'effectue sur la base de transformations inspirées de la génétique, assurant de génération en génération, l'exploration de l'espace des solutions en direction des plus adaptées.
Nous présentons ci-après les concepts, propriétés et heuristiques d'adaptation des AGs ainsi qu'un aperçu des principaux travaux d'application des AGs à la recherche d'information.

3.1. Concepts de base

La transposition des concepts biologiques dans un cadre artificiel a conduit à la définition des concepts suivants :

- *Individu :* structure fondamentale permettant d'encoder une solution candidate au problème posé.

- *Population :* ensemble d'individus d'une même génération.

- *Fonction d'adaptation :* mesure d'efficacité des individus solutions, régissant les transformations génétiques appliquées.

- *Opérateurs génétiques :* procédures de transformation des individus entre deux générations. Les AGs exploitent principalement trois types d'opérateurs :

    - *Sélection :* opérateur de clonage qui respecte le principe de générer une descendance plus nombreuse pour les individus de meilleure valeur d'adaptation. C'est un opérateur orienté vers l'exploitation des individus solutions.

    - *Croisement :* opérateur de combinaison qui agit par paire d'individus en définissant généralement un ou plusieurs sites de coupure. C'est un opérateur appliqué avec une probabilité *Pc ;* il est d'avantage orienté vers l'exploitation des solutions en cours.

    - *Mutation :* opérateur de modification de structure d'individus. C'est un opérateur appliqué avec une probabilité *Pm ;* il est d'avantage orienté vers l'exploration de l'espace de recherche.

3.2. Propriétés et heuristiques d'adaptation

Les principales propriétés d'un AG sont les suivantes :

- *Parallélisme implicite :* en manipulant une population de taille *N*, un AG traite efficacement un nombre de directions de recherche de l'ordre de N^3. Ce résultat dû à Holland [18], traduit la propriété fondamentale des AG's, connue sous le qualificatif de parallélisme implicite. Ceci justifie l'application des AG's à des problèmes d'optimisation caractérisés par des espaces de recherche larges [26].

- *Equilibre entre exploitation et exploration :* l'algorithmique génétique pallie au problème qui résiste depuis longtemps aux méthodes de programmation classique : la détermination d'un équilibre entre l'exploration et l'exploitation [19]. Le mot équilibre est justifié par le fait que les deux procédures sont antagonistes. L'exploitation d'une direction de recherche consiste essentiellement à encourager l'apparition de ses représentants dans la population tandis que l'exploration plaide en faveur de nouvelles directions de recherche. L'AG apporte une solution à ce dilemme en allouant un nombre exponentiellement croissant à la meilleure direction observée [13].

Par ailleurs, de nombreuses heuristiques ont été mises en œuvre dans le but de réguler l'évolution d'un AG en adaptant ses éléments à la nature du problème posé. Nous décrivons en particulier l'heuristique de nichage et des opérateurs augmentés par la connaissance.

- *Nichage* : c'est une heuristique utilisée pour l'ajustement de la fonction d'adaptation dans le cas d'un problème d'optimisation multimodal (présence de plusieurs optimums). A cet effet, l'idée est d'organiser la population en sous-populations, appelées niches, et d'orienter l'exploration de l'espace de recherche

dans les différentes directions qu'elles définissent. Ceci conduit à la découverte et conservation de plusieurs solutions optimales.
D'autres techniques pour la résolution de problèmes multimodaux sont présentés dans [29] [1].

- *Opérateurs augmentés par la connaissance :* l'idée est d'intégrer aux opérateurs génétiques classiques, une connaissance issue du domaine d'application. Le principal intérêt de cette heuristique est d'améliorer la convergence de l'AG en qualité et temps.

3.3. Application à la recherche d'information

Vu sous l'angle de l'optimisation, les techniques de recherche d'information ciblent trois principaux objectifs :

- ***Représentation optimale des documents*** : consiste à couvrir de manière fidèle la sémantique véhiculée par un document en considérant le contenu de la collection

- ***Représentation optimale des requêtes :*** consiste à traduire l'intégralité de la sémantique véhiculée par la requête en considérant le véritable besoin en information de l'utilisateur ainsi que le contenu de la collection interrogée.

- ***Formalisation optimale de la fonction de pertinence :*** cette dernière traduit une combinaison formelle de critères permettant d'estimer la pertinence d'un document relativement à une requête.

Les principales motivations pour la mise en œuvre d'un processus génétique de reformulation de requête sont les suivantes [39] :

- Le fond documentaire peut être perçu comme un espace de dimension élevée. La recherche de requêtes optimales permettant de capturer des voisinages de documents pertinents à la requête utilisateur, évoque à plus d'un titre la puissante capacité d'exploration des AGs.

- Par opposition aux modèles classiques qui focalisent la recherche d'information sur une unique requête, l'AG manipule une population de requêtes dont chacune d'elles peut être à l'origine de la restitution de documents pertinents.
A titre illustratif, nous avons calculé le nombre de documents pertinents qui n'ont aucun terme commun avec la requête initiale ni avec celle déduite par modification pour les topics *350-450* sur la collection adhoc8 (CD4 et CD5) de TREC. On a constaté qu'il y a *1* requête qui n'a aucun terme commun avec *50%* des documents pertinents, *7* requêtes avec *20%* et *16* requêtes avec *10%* . Ceci encourage alors l'idée d'une recherche multi-requêtes.

- La reformulation de requête telle que préconisée dans le modèle vectoriel, manipule les termes indépendamment les uns des autres. Or la pratique a montré que les termes apparaissent dans les documents par combinaison. L'AG apporterait dans ce cas précis, une contribution considérable pour la préservation de « briques élémentaires » qui constituent, dans notre cas, des contextes sémantiques pertinents pour la requête en cours.

Les principaux travaux d'application des AGs à la recherche d'information portent sur la description optimale des documents [14] , optimisation de requête [45] [20] [23] et recherche optimale sur l'Internet [28]. Gordon propose une méthode adaptative de description des documents basée sur les AGs. A chaque document sont associées N descriptions dont chacune est définie par une liste de termes d'indexation [14]. L'application des opérateurs génétiques d'une part, et exploitation du jugement de pertinence d'autre part, font converger une population initiale de descripteurs vers la description optimale du document. Les expérimentations font état d'un accroissement des performances évalué à *25%* à la *40ème* génération de

descripteurs. Gordon exploite ces premiers résultats pour définir un mécanisme de classification des documents [15] basé sur le regroupement de documents pertinents à une même requête. L'expérimentation révèle que la description « génétique » des documents permet d'atteindre *39,74%* d'accroissement des performances au bout de *20* générations et *56.61%* au bout de *40* générations

Yang et Korfhage ont développé un AG pour une optimisation de requête par ré-estimation des poids d'indexation sans induire une expansion [45]. Les générations de populations de requêtes sont renouvelées par application d'une sélection basée sur un échantillonnage stochastique, d'un croisement classique à deux points de coupure et d'une mutation classique. Les expérimentations sur une large collection de documents, révèlent une convergence des variantes de requêtes au bout de *6* générations.

Chen a proposé le système GANNET, basé sur les réseaux de neurones et les AGs [10]. Le processus de recherche d'information est cyclique, opérant en trois phases : phase d'optimisation de concepts, phase d'exploration de concepts et phase de sélection. L'évaluation de GANNET sur une collection moyenne de documents révèle que les performances du système sont très dépendantes des résultats de la recherche d'information. On enregistre une variation d'accroissement des performances de *7%* à *48%* en fonction du nombre de cycles de recherche.

Kraft a appliqué les techniques de programmation génétique dans le but d'effectuer l'optimisation de requêtes booléennes [23]. Les documents sont représentés dans le modèle vectoriel et les requêtes représentées selon le modèle génétique proposé par Koza [21]. Les expérimentations ont montré la faisabilité d'application des techniques de programmation génétique pour la génération de requêtes optimales.

Dans le contexte de la recherche interactive d'informations dans le WEB, Menczer et Belew [28] proposent une méthode génétique de recherche basée sur la coopération d'agents qui effectuent une recherche à contexte local. L'objectif est alors de mettre en œuvre une population d'agents de recherche qui naviguent à travers le réseau d'informations. Ces agents évoluent selon un algorithme évolutif qui optimise la pertinence supposée des documents visités, en réduisant les coûts de recherche. Les expérimentations réalisées montrent globalement l'intérêt de l'approche.

4. Notre approche : un processus génétique spécifique à la reformulation de requête

Le processus de recherche d'information que nous proposons est essentiellement basé sur le déroulement d'un AG qui par essence est cyclique, et vise dans notre cas l'optimisation de requête. Celle-ci consiste à construire de génération en génération, la (les) requête(s) permettant de rappeler le maximum de documents pertinents associés au besoin en information exprimé par l'utilisateur. Ce processus coordonne les activités de l'utilisateur, d'un modèle de recherche de base et d'un AG. Comparativement aux autres travaux du domaine, notre approche est caractérisée par l'application :

- d'une fonction d'adaptation ajustée par le nichage : ceci permet le rappel de documents pertinents pour une même requête mais possédant des descripteurs relativement différents,

- d'opérateurs génétiques non classiques, augmentés par une connaissance théoriquement et expérimentalement approuvée dans le domaine de la reformulation de requête par injection de pertinence : ceci permet d'accélérer le processus de recherche d'information par une exploration guidée de l'espace des documents.

4.1. Le processus d'optimisation de requête

Le processus général d'optimisation de requête, illustré sur la figure 1, est fondé sur l'évolution génétique de niches de requêtes. Une niche est un groupe potentiel de requêtes qui investit une direction de recherche déterminée et évolue en accord avec les résultats d'évaluation de la recherche, traduit par deux facteurs : valeur d'adaptation relative des niches de requêtes, jugement de pertinence de l'utilisateur.

4.2. Eléments de l'AG

Cette section présente les éléments caractéristiques de l'AG d'optimisation de requête que nous proposons.

4.2.1. Individu, Niche et Population

- *Individu :* un individu requête est représenté comme suit :

$$\begin{array}{cccc} & t_1 & t_2 & t_T \\ Q_u^{(s)} & (q_{u1} & q_{u2} & q_{uT}) \end{array}$$

Où :

$Q_u^{(s)}$: Individu requête n° u de la population à la génération s

$t_1, t_2, ..., t_T$: Liste de termes d'indexation

q_{ui} : Poids du terme t_i dans la requête individu $Q_u^{(s)}$

T : Nombre total de termes d'indexation dans la collection

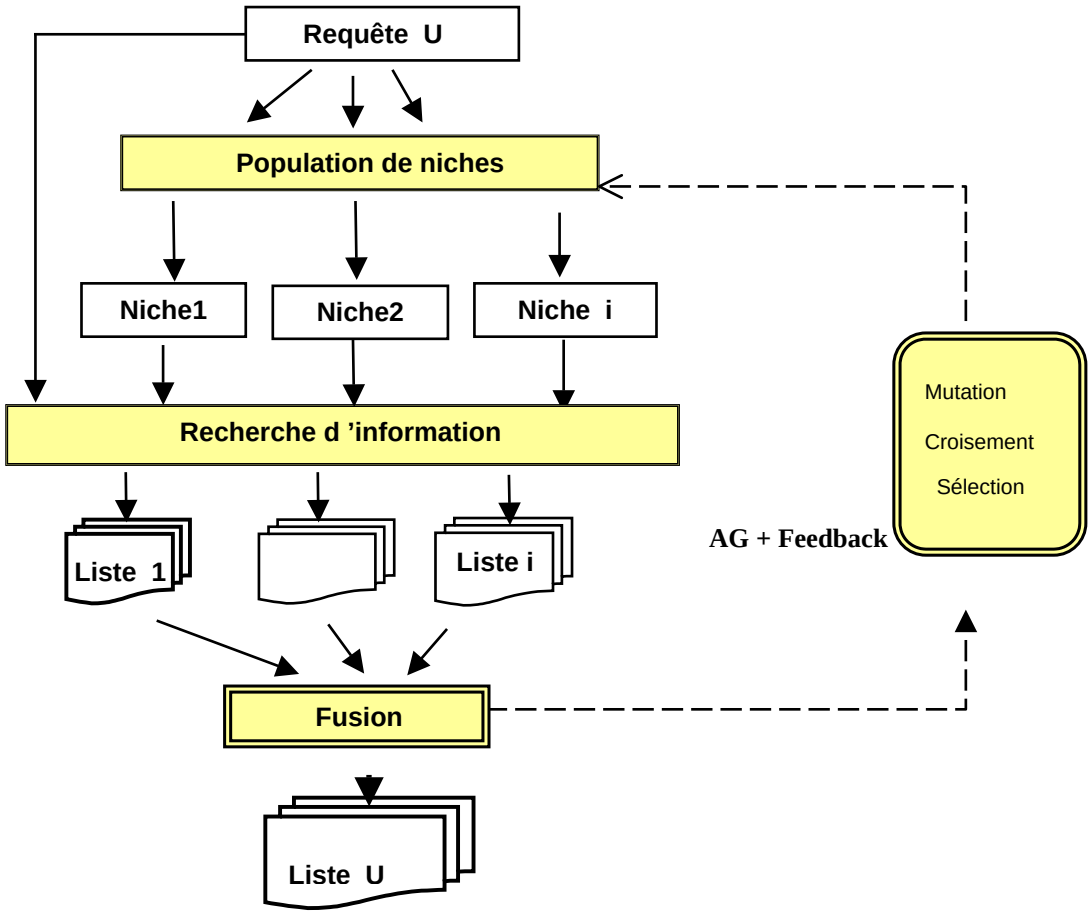

FIG. 1 : *Le processus génétique d'optimisation de requête*

- *Niche :* la technique de nichage est intégrée à un AG en vue de résoudre un problème d'optimisation multimodal. Dans notre cadre de travail, nous considérons que la fonction de pertinence est multimodale de par la présence éventuelle de documents pertinents à une même requête dans des sous espaces documentaires différents et que nous appelons *régions*. Ces régions sont, à notre sens, structurellement assez dissemblables, pour être atteintes par une « unique requête optimale ». A cet effet, l'AG favorisera la production de niches potentielles de requêtes associées aux différentes régions. C'est ainsi le principe de la ***recherche coopérative.*** On définit la relation **Coniche** notée $\equiv_N$, comme suit :

$$(Q_u^{(s)} \equiv_N Q_v^{(s)}) \Leftrightarrow (|(Ds(Q_u^{(s)}, L) \cap (Ds(Q_v^{(s)}, L) | > Limite\_Coniche)$$

Où :
Ds(Q, L) : Ensemble des L premiers documents sélectionnés par la requête individu Q
Limite_Coniche : Nombre minimal de documents communs retrouvés par les individus requêtes d'une même niche.
Avec :
Limite_Coniche = NbJug*Prop_Coniche
Où :
Prop_Coniche: Constante réelle appartenant à l'intervalle [0 1]
NbJug : Nombre de documents jugés par l'utilisateur

- *Population :* la population est représentée par un ensemble de niches, renouvelée à chaque génération sous l'effet de l'évaluation des individus requêtes et de l'application des opérateurs génétiques. La population initiale est constituée des descripteurs des documents pertinents, complétée par les documents les plus ressemblants situés en début de la liste présentée à l'utilisateur. Précisons qu'un descripteur de document est une liste pondérée de termes d'indexation et peut donc exprimer une requête. En outre, on intègre à chaque prochaine génération de l'AG, l'individu de meilleure valeur d'adaptation ainsi qu'une requête virtuelle constituée des termes les plus fréquents dans les documents pertinents, issus de la génération courante.

4.2.2. Fonction d'adaptation

A chaque individu requête est associé une mesure d'adaptation définie par la formule suivante [40] :

$$Fitness(Q_u^{(s)}) = \frac{\sum_{dr \in Dr(s), dnr \in Dnr(s)} J(Q_u^{(s)}, dr) - J(Q_u^{(s)}, dnr)}{\sum_{dr \in Dr(s), dnr \in Dnr(s)} \left| J(Q_u^{(s)}, dr) - J(Q_u^{(s)}, dnr) \right|}$$

Où :
J : Mesure de Jaccard
$Dr^{(s)}$: Documents pertinents retrouvés à la génération s de l'AG
$Dnr^{(s)}$: Documents non pertinents retrouvés à la génération s de l'AG
dr : Document pertinent
dnr : Document non pertinent
Avec :

$$J(Q_u^{(s)}, D) = \frac{\sum_{i=1}^{T} q_{ui}^{(s)} d_i}{\sum_{i=1}^{T} q_{ui}^{(s)2} + \sum_{i=1}^{T} d_i^2 - \sum_{i=1}^{T} q_{ui}^{(s)} d_i}$$

Où :
$q_{ui}^{(s)}$: Poids d'indexation du terme t_i dans la requête $Q_u^{(s)}$
d_i : Poids d'indexation du terme t_i dans le document D

Le principal avantage de ce type de fonctions est qu'il est basé sur le jugement de pertinence des utilisateurs et sur un modèle de fonction statistiquement corrélé aux mesures de taux de rappel/ précision [5].

4.2.3. Opérateurs génétiques

Les opérateurs génétiques que nous proposons ne sont pas classiques; ils sont augmentés par des techniques d'expansion et repondération de requête définies dans les méthodes de reformulation de requête par injection de pertinence. L'utilisation d'une connaissance auxiliaire propre au problème de la recherche d'information permettrait d'accélérer l'exploration génétique par une recherche guidée dans l'espace des documents.
L'application de ces opérateurs s'effectue de manière restreinte aux niches et non de manière uniforme sur toute la population.

- *Sélection :* nous optons pour une sélection basée sur le mode d'espérance mathématique [13]. Ce mode de sélection consiste essentiellement à générer pour chaque individu de la population, un nombre de clones dépendant directement de sa valeur d'adaptation.

- *Croisement :* nous proposons un croisement basé sur le poids des termes, qui traduit une expansion et repondération contextuelles de requête. Le principe de ce croisement est défini comme suit :

$$\left.\begin{array}{l} Q_u^{(s)}\,(\,q_{u1}^{(s)}, q_{u2}^{(s)}, \ldots, q_{uT}^{(s)}) \\ Q_v^{(s)}\,(\,q_{v1}^{(s)}, q_{v2}^{(s)}, \ldots, q_{vT}^{(s)}) \end{array}\right\} \longrightarrow \quad Q_p^{(s+1)}\,(\,q_{p1}^{(s+1)}, q_{p2}^{(s+1)}, \ldots, q_{pT}^{(s+1)})$$

Si $((q_{ui}^{(s)} \neq 0) \wedge (q_{vi}^{(s)} \neq 0))$ *Alors*
$q_{pi}^{(S+1)} =$ *Max* $(q_{ui}^{(s)}, q_{vi}^{(s)})$ *si Poids* $(t_i, Dr^{(s)}) \geq$ *Poids* $(t_i, D_{nr}^{(s)})$
Min $(q_{ui}^{(s)}, q_{vi}^{(s)})$ *sinon*
Sinon
Si $(q_{ui}^{(s)} = 0)$ *Alors*
$q_{pi}^{(s+1)} = q_{vi}^{(s)}$
Sinon
$q_{pi}^{(s+1)} = q_{ui}^{(s)}$
Fsi
Fsi

Où :
$Poids(t_i, D) = \sum_{dj \in D} d_{ji}$, D : Ensemble de documents, d_{ji} : poids du terme t_i dans le document d_j

- *Mutation :* cet opérateur, basé sur la pertinence des termes, consiste essentiellement à exploiter les termes présents dans les documents pertinents dans le but d'ajuster les valeurs des gènes correspondants dans les requêtes sélectionnées pour la mutation. Plus précisément, la mutation est effectuée selon l'algorithme suivant :

Début
Pour *chaque* t_i *dans la liste Lmut* ***Faire***
Si *(Random(p)<Pm)* ***Alors***
$q_{ui}^{(s)} = Poids\_Moyen(Q_u^{(s)})$-$\delta$
Finsi
Fait
Fin

Où :
Random(p) : Fonction qui génére un nombre aléatoire p dans l'intervalle [0 1]

$$Poids\_Moyen(Q_u^{(s)}) = \frac{\sum_{j=1}^{T} q_{ui}^{(s)}}{n_{q_{ui}^{(s)}}}$$

$n_{qui}^{(s)}$: Taille effective de l'individu requête $Q_u^{(s)}$ (nombre de termes d'indexation de poids non nuls)
δ : Paramètre de contrôle du poids moyen
Lmut : liste de nouveaux termes issus des documents pertinents et trié selon le score calculé comme suit :

$$Score(t_i) = \frac{\sum_{d \in Dr^{(s)}} d_i^{(s)}}{\left| Dr^{(s)} \right|}$$

En conséquence, la probabilité effective de mutation d'un terme varie en fonction de sa distribution dans les documents pertinents. Ceci constitue alors, à notre sens, une variante de la mutation auto-adaptative [3].

4.3. Principe de fusion des résultats de recherche

La population d'individus requêtes est organisée en niches. A l'issue de l'évaluation de chaque niche, nous obtenons des ensembles non disjoints de documents restitués par le processus de recherche de base. Ces listes partielles sont fusionnées de manière à constituer une liste unique de documents, soumise au jugement de pertinence de l'utilisateur. Le principe adopté pour la fusion a ainsi un impact direct sur la précision de la recherche.
Dans ce cadre, nous proposons une fusion basée sur l'ordre global de la population. On propose une **fusion sélective** qui consiste à fusionner linéairement les documents restitués par les requêtes dont les valeurs d'adaptation sont supérieures à la moyenne d'adaptation dans la population. L'ordre des documents restitués à l'utilisateur est obtenu en calculant :

$$Rel(D_j) = \sum_{N_j^{(s)} \in Pop^{(s)}} \sum_{Q_u^{(s)} \in N_j^{(s)}} Fitness(Q_u^{(s)**} * RSV(Q_u^{(s)}, D_j))$$

Où :
$Q_u^{(S)**}$: Requêtes dont la valeur d'adaptation est supérieure à l'adaptation moyenne de la population
RSV (Q,D) : Valeur de pertinence calculée du document D relativement à la requête Q
$N_j^{(s)}$: Niche j de la population à la génération s de l'AG

On note ainsi que c'est la valeur d'adaptation directe de chaque individu requête qui contribue à déterminer la qualité de l'ordre des documents restitués.

5. Validation de notre approche

Nous présentons dans cette section les différentes expérimentations que nous avons réalisées à l'aide du SRI Mercure, dans le but de valider notre approche. Plus précisément, notre objectif est d'évaluer :

- l'apport de notre approche d'optimisation de requête pour l'amélioration des performances du SRI. Cette évaluation est effectuée en utilisant des estimations comparatives des nombres de documents pertinents restitués par itération ainsi que nombre de documents pertinents cumulé et ce, comparativement à une recherche de base menée dans le SRI Mercure,
- l'impact des éléments de l'AG sur les résultats de recherche.

5.1. Méthode d'évaluation

La mesure de performances du processus de recherche d'information a été effectuée par la méthode d'évaluation de la collection résiduelle. Cette méthode est adoptée afin d'évaluer le processus d'optimisation de requête induit par l'AG à des itérations feedback successives correspondant à des générations de l'algorithme.
Nous avons, à cet effet, mis en œuvre la base de référence d'évaluation notée *Baseline*, qui considère comme référence, à chaque génération *s* de l'AG, l'apport de nouveaux documents relativement à la génération *s-1*.
Ce type d'évaluation mesure ainsi la capacité de l'AG à atteindre de nouveaux documents pertinents lors de l'évaluation de chaque nouvelle génération de requêtes. Les bases de test utilisées sont des sous-collection de la base TREC pour Text Retrieval Conference [42].
On calcule pour chaque requête de la base de test, la précision exacte à 15 documents puis le nombre de documents pertinents restitués et nombre de documents pertinents cumulé à chaque itération et ce, sur l'ensemble des requêtes de la base. En outre, l'accroissement des performances est calculé sur les bases suivantes :

- une liste de ***15*** documents présentés et soumis au jugement de l'utilisateur : cette valeur est communément utilisée pour l'évaluation des techniques de recherche d'information [33] [17] [6],

- une série de *5* itérations feedback,

- une recherche initiale se limitant à l'interrogation du système Mercure avec la requête utilisateur.

5.2. Résultats préliminaires

Nous avons en premier lieu évalué globalement la faisabilité de notre approche en utilisant : une fonction de nichage basée sur une distance seuillée entre individus requêtes, une fonction d'adaptation basée sur la similitude avec les documents pertinents et des opérateurs génétiques appliqués uniformément sur l'ensemble de la population.
Les expérimentations réalisées sur la collection TREC6 French Data contenant *141490* documents avec *21* requêtes nous ont permis d'aboutir aux principaux résultats suivants [7] :
- les probabilités de croisement *Pc* et mutation *Pm* ont un impact considérable sur les résultats de la recherche avec, cependant, un degré plus important pour le croisement. Les valeurs des probabilités donnant les meilleurs résultats sont *Pc=0.7* et *Pm=0.07*,
- l'optimisation génétique de requêtes donne de meilleurs résultats de recherche à partir d'une taille de population égale à *2* et ce, à chaque itération feedback,
- l'application d'opérateurs génétiques augmentés assure un taux d'accroissement des performances de *97%* à *255%* comparativement à l'application d'opérateurs classiques.

5.3. Evaluation globale

Nous décrivons à présent les expérimentations réalisées pour une évaluation globale de notre approche. Ces expérimentations sont réalisées sur la collection de test AP88 contenant 144186 documents et basée sur l'algorithme de recherche suivant :

1. Iter :=*0*
2. Présenter puis évaluer la requête initiale
3. Juger les *15* premiers documents
4. Construire la population initiale de niches de requêtes
5. Pour chaque niche
6. Effectuer la recherche pour chaque individu requête
7. Fait

8. Fusionner les résultats
9. Construire les nouvelles niches
10. Juger les *15* premiers documents
11. Pour chaque niche
12. Calculer l'adaptation pour chaque individu requête
13. Appliquer les opérateurs génétiques
14. Fait
15. Iter :=Iter+1
16. Si (Iter<5) Alors aller à 4

5.3.1. Impact de la taille de population et structure de la population

Notre approche d'optimisation de requête s'articule sur une recherche coopérative menée par une population de niches de requêtes. Nous évaluons alors, en premier lieu, la taille de la population et seuil de conichage qui est un facteur déterminant quant à l'organisation de la population en niches. Les résultats obtenus en termes de nombre de documents pertinent et documents pertinents cumulé (nombre entre parenthèses) à chaque itération, sont présentés sur le tableau 1.

Nous constatons que la proportion de conichage a un effet intrinsèque sur les résultats de recherche et effet combiné à celui de la taille de population. La comparaison du nombre de documents pertinents cumulé obtenu à la cinquième itération, nous permet de déterminer les meilleures paires de valeurs des paramètres *Taille_Pop* et *Prop_*Coniche et qui sont respectivement *(2, 0.2), (4, 0.6)* et *(6, 0.2).*

Par ailleurs, l'estimation du nombre de niches de requêtes construites à chaque génération, pour les différentes valeurs de la taille de population et proportion de conichage, nous a permis de constater une dépendance linéaire entre ces paramètres. Plus précisément, le nombre de niches varie dans l'intervalle *[1 Taille_Pop]* avec un écart plus important pour des valeurs relativement élevées de la proportion de conichage.

Prop_Coniche	**Iter1**	**Iter2**	**Iter3**	**Iter4**	**Iter5**
			Taille_Pop=2		
0.2	172(172)	113(285)	87(372)	80(452)	**70(522)**
0.6	172(172)	113(285)	87(372)	75(447)	71(518)
1	172(172)	113(285)	89(374)	69(443)	69(513)
			Taille_Pop=4		
0.2	180(180)	88(268)	93(361)	87(448)	61(509)
0.6	180(180)	88(268)	98(366)	75(442)	**78(520)**
1	180(180)	88(268)	97(365)	75(440)	57(497)
			Taille_Pop=6		
0.2	177(177)	105(282)	80(362)	61(423)	**68(491)**
0.6	177(177)	105(282)	78(360)	64(424)	56(480)
1	177(177)	105(282)	60(342)	68(410)	50(460)

Tableau 1. : Effet de la taille de population et proportion de conichage

Ces résultats nous amènent à conclure que :

1. L'augmentation de la taille de population augmente le risque de construction de requêtes dissemblables sous l'effet combiné du croisement et de la mutation et par conséquent, une croissance du nombre de niches.

2. L'augmentation de la proportion de conichage conduit à l'imposition d'une condition plus stricte quant au « rapprochement » des résultats d'évaluation des individus requêtes devant appartenir à la même niche. Ceci a pour conséquence, l'apparition de nouvelles niches qui explorent de nouveaux voisinages documentaires.

L'analyse de cette série d'expérimentations est complétée par l'examen des valeurs de précision moyenne *(Pmoy)* et précision à 15 documents *(P15)* obtenues et présentées sur le tableau 2.
Notons, à l'aide des résultats mis en gras, et en privilégiant les précisions obtenues aux premières itérations, que la meilleure valeur de taille de population varie entre *2* et *4.*
En faisant un compromis entre les résultats obtenus en utilisant la mesure du nombre de documents pertinents et ceux obtenus en utilisant la mesure de précision, nous convenons de retenir pour les expérimentations ultérieures une taille de population de *4* avec la valeur médiane de *0.6* pour la proportion de conichage.

Prop_Coniche	**Iter1**		**Iter2**		**Iter3**		**Iter4**		**Iter5**	
	Pmoy	***P15***	***Pmoy***	***P15***	***Pmoy***	***P15***	***Pmoy***	***P15***	***Pmoy***	***P15***
					Taille_Pop=2					
0.2	0.20	0.47	**0.10**	**0.31**	0.07	0.24	0.05	0.22	0.03	0.19
0.6	0.20	0.47	**0.10**	**0.31**	0.07	0.24	0.05	0.20	0.03	0.19
1	0.20	0.47	**0.10**	**0.31**	0.07	0.24	0.05	0.19	0.04	0.19
					Taille_Pop=4					
0.2	**0.21**	**0.51**	0.07	0.24	**0.06**	**0.25**	0.06	0.24	0.03	0.16
0.6	**0.21**	**0.51**	0.07	0.24	**0.06**	**0.27**	0.06	0.21	0.04	0.21
1	**0.21**	**0.51**	0.07	0.24	**0.06**	**0.26**	0.05	0.20	0.03	0.15
					Taille_Pop=6					
0.2	0.22	0.49	0.09	0.29	0.05	0.22	0.04	0.16	0.03	0.18
0.6	0.22	0.49	0.9	0.29	0.05	0.21	0.04	0.17	0.03	0.15
1	0.22	0.49	0.09	0.29	0.04	0.16	0.04	0.18	0.03	0.13

Tableau 2 : Variation de la précision en fonction de la taille de population et proportion de conichage

5.3.2. Impact de l'optimisation génétique de requête

Nous nous intéressons à présent à l'évaluation de l'impact du processus génétique d'optimisation de requête, notée *Avec_AG*, sur les résultats de recherche. L'évaluation est basée sur l'estimation du nombre de documents pertinents et nombre de documents pertinents cumulé à chaque itération et ce, comparativement à une exécution *Sans_ AG* qui considère les résultats de l'itération précédente. Les résultats obtenus sont présentés sur le tableau 3. La ligne *Accroissement/Doc_Pert_Cum* exprime le taux d'accroissement enregistré en nombre de documents pertinents cumulé, à chaque itération, entre une exécution du processus de recherche *Avec_AG* et autre exécution *Sans_AG*.

	Iter1	**Iter2**	**Iter3**	**Iter4**	**Iter5**
Avec_AG	180(180)	88(268)	97(366)	75(442)	78(520)
Sans_AG	110(110)	114(225)	77(302)	69(371)	65(437)
Accroissement/Doc_Pert_Cum	63%	18%	21%	22%	22%

Tableau 3 : Comparaison entre nombres de documents pertinents retrouvés

Les résultats montrent que la recherche par optimisation génétique de requête assure, à chaque itération, un accroissement des performances qui varie de *18%* à *63%*. Notons cependant, un pic de performances à la première itération puis diminution de l'accroissement des performances au niveau de la deuxième itération puis reprise aux itérations suivantes. Ceci peut être justifié par la conjonction de deux causes. La première, concerne le principe adopté pour la construction de la population initiale. Cette dernière est en effet, constituée d'une niche composée d'individus requêtes sélectionnés dans le voisinage des documents jugés pertinents à la recherche initiale, ce qui nous permet d'atteindre immédiatement de nouveaux documents pertinents ressemblants.
La deuxième cause est liée à la première. En effet, en raison du rappel d'un nombre important de documents pertinents à la première itération, les itérations suivantes révéleront des performances moindres, eu égard au nombre de documents pertinents total limité d'une part, et restriction de l'espace documentaire exploré relativement aux niches en cours d'évolution, d'autre part.

5.3.3. Impact des opérateurs génétiques augmentés

Les opérateurs génétiques proposés dans notre approche, sont augmentés par une connaissance liée aux techniques de reformulation de requête par injection de pertinence. Nous évaluons à présent, l'impact de ces opérateurs spécifiques sur les résultats de recherche. A cet effet, nous avons effectué une série d'expérimentations en utilisant les opérateurs préalablement définis et comparé les résultats à ceux obtenus par application d'opérateurs génétiques classiques .
Le tableau 4 présente le nombre de documents pertinents par itération et nombre de documents pertinents cumulé obtenus par application de chacun de ces groupes d'opérateurs. La ligne *Accroissement* précise le taux d'accroissement du nombre de documents pertinents, enregistré par application des opérateurs augmentés et ce, comparativement à l'application d'opérateurs classiques.

	Iter1	**Iter2**	**Iter3**	**Iter4**	**Iter5**
Opérateurs classiques	171(171)	79(250)	65(315)	65(380)	68(449)
Opérateurs augmentés	180(180)	88(268)	97(366)	75(442)	78(520)
Accroissement	5,2%(5,2%)	5,2%(7,2%)	4,9%(16%)	15%(16%)	14%(15%)

Tableau 4 : Impact des différents jeux d'opérateurs génétiques sur les résultats de recherche

On constate clairement que les opérateurs génétiques augmentés assurent de meilleurs résultats de recherche. Plus précisément, on enregistre un accroissement de *5%* à *15%* du nombre de documents pertinents cumulés à chaque itération. Ceci confirme l'intérêt de l'intégration de la connaissance spécifique aux techniques de recherche d'information, à la structure des opérateurs.

6. Conclusion

Nous avons présenté dans cet article, une approche basée sur l'exploitation conjointe des techniques d'algorithmique génétique et de reformulation de requête par injection de pertinence. La population est manipulée en niches constituant des sous populations qui explorent simultanément des régions différentes de l'espace documentaire. L'AG standard opère alors sur chaque niche et exploite les résultats de sa recherche en accord avec sa valeur d'adaptation relative dans la population.
Les opérateurs génétiques proposés sont augmentés par une connaissance liée aux techniques de reformulation de requête et, adaptés à l'évolution de niches de requêtes.
Les expérimentations et évaluations que nous avons réalisées confirment l'intérêt de notre approche. Nous avons en particulier étudié l'apport de performances dû aux éléments de l'AG d'optimisation de requête que nous proposons et évalué l'impact de chacun de ses paramètres sur les résultats de recherche.
Une première série d'expérimentations nous a permis d'estimer l'effet combiné de la taille de population et proportion de conichage. On a alors montré la nécessité d'ajuster ces paramètres afin d'assurer un équilibre

entre nombre de requêtes et facteur de multiplicité des directions de recherche. L'optimisation génétique de requête assure un accroissement des performances de *18%* à *63%* relativement à la référence et ce, en fonction des itérations feedback. Nous avons également procédé à l'évaluation de l'impact des opérateurs génétiques proposés. Il en ressort principalement que l'intégration de la connaissance à la structure des opérateurs génétiques accroît les performances de recherche.

Nous concluons ainsi sur la faisabilité pratique d'adaptation d'un AG classique au contexte de la reformulation de requête. Nous envisageons dans un proche avenir, d'intégrer à l'AG proposé diverses techniques avancées. Dans ce sens, nos perspectives s'inscrivent principalement dans trois volets.

Le premier concerne la révision du principe d'application des opérateurs génétiques. Plus précisément, nous nous orientons vers une application restrictive du croisement [29] et exploitation du principe de mutation auto-adaptative par contrôle de la probabilité de mutation en cours de recherche [3][4]. Cette adaptation sera régulée d'une part par la distribution des termes dans les documents pertinents et d'autre part, par la densité des régions de pertinence déterminées par les niches de requêtes.

Dans le second volet, l'extension de l'AG consisterait à exploiter la technique de *clearing* proposée par Petrowski [29] qui nous conduirait à la préservation des meilleures requêtes de chaque niche et ainsi se prémunir des effets éventuels de dispersion dans l'espace de recherche due à l'application des opérateurs génétiques d'une part et de la fusion d'autre part.

Enfin, il serait intéressant dans un troisième volet, d'exploiter des éléments de la théorie de l'optimisation subjective [41] de manière à inscrire le processus de recherche génétique dans un cadre plus adapté à la coopération utilisateur – SRI.

Actuellement, nous poursuivons ce travail en nous intéressant plus particulièrement à l'adaptation de cette approche à une reformulation automatique de requête. Il convient alors de déterminer une base de supposition de pertinence d'une part et seuil de convergence de l'AG d'autre part. Ceci permettrait de généraliser notre approche à divers modes d'interrogation du SRI.

Références

[1] Andrew H., Watson and Ian C. Parmee. Steady state genetic programming with constrained complexity crossover using species and subpopulations. *International Conference on Genetic Algorithms, ICGA'97* july 19-23, 1997.

[2] R. Attar, S. Franenckel. Local Feedback in Full Text Retrieval Systems. *Journal of the ACM,* pages 397-417, 1977.

[3] Th. Back. Self adaptation in genetic algorithms. F.J Varela and P. Bourgine Editors : Proceedings of the 1st Europeen Conference on Artificial Life, pp 263-271, The MIT Press, Cambridge MA, 1992.

[4] TH. Back and M. Schutz.. Intelligent mutation rate control in canonical genetic algorithms. W. Ras and Michalewicz Editors : Foundation of Intelligent Systems 9 th International Symposium, ISMIS'96, pp 158-167, Springer, Berlin, 1996.

[5] B. Bartell, G. Cottrel & R.K Belew. Automatic Combination of Multiple Ranked Retrieval Systems. *In Proceedings of the ACM SIGIR, Conference on Research and Development in Information Retrieval, 1994, pages 173-181*

[6] M. Boughanem, C. SouleDupuy. Mercure at TREC 6. *In Harman DK, ed. 6th International Conference on Text Retrieval TREC 6.* November 21-23, NIST SP, pages 321-328, 1997.

[7] M. Boughanem, C. Soule-Dupuy. Query Modification Based on Relevance Back-propagation in Adhoc Environment. *Information Processing and Management,* volume 35 n°2, pages 121-139, 1999.

[8] M. Boughanem, C. Chrisment & L.Tamine Genetic Approach to Query Space Exploration. *Information Retrieval Journal* volume 1 N°3 , pages 175-192, 1999.

[9] C. Buckley, G. Salton & J. Allan The Effect of Adding Information in a Relevance Feedback Environment, *In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR),* pages 292-300,1994

[10] H. Chen. Machine Learning for Information Retrieval Neural Networks, Symbolic Learning and Genetic Algorithms, *JASIS,* 46(3). pages 194-216, 1995.

[11] S. Dumais. Latent Semantic Indexing (LSI), TREC3 report. *In Proceedings of the 3rd Conference on Text Retrieval Conference,* pages 105-115, 1994

[12] Fogel L., Owens A. J., Walsh M. J. Artificial Intelligence through Simulated Evolution. *New York, Jhon Wiley,* 1996

[13] Goldberg D.E. Algorithmes Génétiques, Exploration, Optimisation et Apprentissage Automatique. *Edition Addison Wesley*, 1994

[14] M. Gordon. Probabilistic and Genetic Algorithms for Document Retrieval. *Communications of the ACM* pages 1208-1218, October 1988

[15] M. D. Gordon. User-Based Document Clustering By Redescribing Subject Descriptions with a Genetic Algorithm. *Journal of The American Society for Information Science*, 42(5) pages 311 - 322, 1991

[16] D. Haines & W.B Croft. Relevance Feedback and Inference Networks. In the proceedings of the ACM SIGIR, *Conference on Research and Development in Information Retrieval* , pages 2-11, 1993

[17] D. Harman. Relevance :Feedabck Revisited. In Proceedings of the ACM SIGIR *Conference on Research and Development in Information Retrieval (SIGIR),* pages1-10, 1992

[18] Holland J. Adaptation In Natural and Artificial Systems. *University of Michigan Press,* Ann Arbor, 1975

[19] Holland J.. Les Algorithmes Génétique. *Revue POUR LA SCIENCE* N°=179 pages 44-51, Septembre 1992

[20] H. Jing & E. Tzoukermann. Information Retrieval Based on Context Distance and Morphology. *In proceedings of the ACM SIGIR, Conference on Research and Development in Information Retrieval*, pages 90-96, August, 1999, Buckley USA

[21] Koza J.R: A Hierachical Approach to Learning the Boolean Multiplexer Function. *In Rawlins G. Ed., Foundations of Genetic Algorithms, Morgan Kaufman*, San Mateo, CA, pages 171-192, 1991

[22] Koza J. R. Genetic Programming. *Bradford book, MIT Press*, Cambridge, MA, USA 1992.

[23] Kraft DH, Petry FE, Buckles BP and Sadisavan T. Applying Genetic Algorithms to Information Rtrieval System Via Relevance Feedback. *In Bosc and Kacprzyk J eds, Fuzziness in Databse Management Systems Studies in Fuzziness* Series, Physica Verlag, Heidelberg, Germany pages 330-344

[24] Kwok K. L. A neural network for probabilistic information retrieval. *In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval,* , pages 21-30, 1989.

[25] R. Mandala, T.Tokunaga & H. Takana, Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion. *In proceedings of the ACM SIGIR, Conference on Research and Development in Information Retrieval*, pages 191-197, August, 1999, Buckley USA

[26] Z. Michalewicz. Genetic algorithms + Data structures = Evolutionary programs, Springer Verlag, New York 3rd Edition, 1996

[27] M. Mitra, A. Singhal, C.Buckley. Improving Automatic Query Expansion. *In proceedings of the ACM SIGIR, Conference on Research and Development in Information Retrieval, pages 206-214, 1998*

[28] F. Menczer , R.K. Belew. Adaptive Retrieval Agents. Internalizing Local Context and Scaling up to the WEB. *Machine Learning*, pages 1-45, Kluwer Academic Publishers, 1999

[29] A. Petrowski. A new selection operator dedicated to speciation. International Conference on Genetic Algorithms *ICGA*, july 19-23, 1997

[30] Y. Qiu & H.P. Frei. Concept Based Query Expansion. *In Proceedings of the 16th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 60-169, Pittsburg, USA 1993

[31] S.E Robertson & K. Sparch Jones. Relevance Weighting for Search Terms. *Journal of The American Society for Information Science*, Vol 27, N°3, pages 129-146, 1976

[32] S.E Robertson, S.E. Walker & M.M Hnacock-Beaulieu. Large Test Collection Experiments on an Operational Interactive System : Okapi at TREC. *Information Processing and Management volume 31, n°3* pages 260-345, 1995

[33] Robertson S., Walker S.. On Relevance Weights with Little Relevance Information. *In Proceedings of the ACM SIGIR Conference on Research and Development*, pages 16-24, 1997

[34] J. J Rocchio. Relevance Feedback in Information Retrieval, in The Smart System Experiments in Automatic Document Processing. *G.Salton, Editor, Prentice-Hall, Inc., Englewood Cliffs*, NJ, pages 313-23, 1971

[35] G. Salton. Automatic Information and Retrieval. *Mcgraw hill Book Company*, N. Y., 1968

[36] G. Salton. Automatic Text Processing. The Transformation Analysis and Retrieval of Information by Computer. *Addison Wesley*, Reading 1989

[37] Schutze H., Pedersen J.. A Cooccurrence- Based Thesaurus and two Applications to Information Retrival. *Information Processing & Management*, 33(3). pages 307-318, 1997

[38] I. Syu & S. D Lang. A Competition-Based Connexionist Model for Information Retrieval. *Intelligent Multimedia Information Systems and Management (RIAO)*, New York, Vol 1. pages 248-265,1994

[39] L. Tamine. Reformulation Automatique de Requête basée sur l'Algorithmique Génétique. *Actes du Congrès Inforsid*, pages 643-662, Toulouse Juin 1997

[40] L. Tamine, M. Boughanem. Query Optimisation Using an Improved GA. *Conference on Information Knowledge and Management CIKM*, pages 368-373Washington, November 2000

[41] G. Venturini, M. Slimane, F. Morin and Asselin Beauville. On using interactive genetic algorithms for knowledge discovery. *International Conference on Genetic Algorithms, ICGA'97* july 19-23, 1997.

[42] E.M. Voorhees, TREC Overview. *In Proceedings of the seventh Text Retrieval Conference TREC7*, 1999

[43] R. Wilkinson & P. Hingston. Using The Cosine Measure in A Neural Network for Document Retrieval. *In the Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 202-210, Chicago (USA), 1991

[44] J. Xu & W.B. Croft. Query Expansion Using Local and Global Document Analysis. *In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval , pages 4-11 ,* Zurich, 1996

[45] J. J Yang & R. R Korfhage Query Optimisation in Information Retrieval Using Genetic Algorithms. *International Conference on Genetic Algorithms ICGA, 1993*