UNIVERSITE KASDI MERBAH OUARGLA
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Informatique et Technologie de l'Information

Mémoire

MASTER ACADEMIQUE
Domaine : Mathématiques et Informatique
Filière : Informatique
Option: Informatique industrielle

Présenté par
GUESSOUM Asma

Thème

EXPANSION DE REQUETE EN UTILISANT L'ALGORITHME DES FEUX D'ARTIFICE

Devant le jury

M.	Mahdjoub Mohammed Bachir	MAA	UKM Ouargla	Président
M.	Mezati Messaoud	MAA	UKM Ouargla	Examinateur
M.	BEKKARI Fouad	MAA	UKM Ouargla	Encadreur

Année universitaire : 2020 /2021

Remerciements

*Je tiens à remercier **le dieu**, le tout puissant de ma donnée la patience, la santé et le courage pour finir ce travail.*

*Je tiens à remercier profondément mon encadreur: Monsieur **BEKKARI FOUAD** qui m'a encouragé pour finir ce travail.*

Dédicaces

C'est avec joie que je dédie ce modeste travail :

A l'âme pure de mon père.

À ma très cher mère qui j'espère rendra fière de moi,

A mon mari Guemmoula Lazhar qui m'a beaucoup aider et grâce à lui que je suis là et à mes enfants que j'aime bien

À toute la famille

À mes amis et mes collègues

Asma

Résumé :

La clarté et la précision de la requête sont un facteur clé de l'efficacité des processus de récupération d'informations et de la qualité des résultats, mais l'utilisateur écrit généralement de courtes requêtes qui affectent la précision des résultats. Plusieurs approches ont été utilisées pour résoudre le problème de l'expansion des requêtes ; qui a démontré une efficacité variable en termes de résultats obtenus.

Par contre, L'optimisation combinatoire est une approche qui connaît un développement scientifique rapide, ce domaine consiste à trouver les meilleures solutions et les meilleurs résultats pour nombre très grands des solutions disponibles.

Nous traiterons le problème de l'expansion des requêtes en tant que problème combinatoire et nous utiliserons l'algorithme des feux d'artifice pour résoudre ce problème, qui s'est avéré efficace pour résoudre bon nombre des problèmes standard.

Mots clés

Recherche d'informations, l'expansion des requêtes, problème combinatoire, algorithme de feux d'artifice

ملخص:

يعد وضوح الاستعلام ودقته عاملاً أساسيًا في كفاءة عملية استرجاع المعلومات وجودة النتائج ، ولكن المستخدم يكتب عادةً استعلامات قصيرة تؤثر على دقة النتائج. تم استخدام عدة طرق لحل مشكلة توسيع الاستعلام ؛ التي أظهرت فعالية متغيرة من حيث النتائج التي تم الحصول عليها.

من ناحية أخرى ، التحسين التوافقي هو نهج يخضع لتطور علمي سريع ، ويعتمد هذا المجال من إيجاد أفضل الحلول وأفضل النتائج لعدد كبير جدًا من الحلول المتاحة.

سنتعامل مع مشكلة توسيع الاستعلام على أنها مشكلة اندماجية وسنستخدم خوارزمية الألعاب النارية لحل هذه المشكلة ، والتي ثبت أنها فعالة في حل العديد من المشكلات القياسية.

الكلمات المفتاحية:

البحث عن المعلومات ، توسيع الاستعلام ، المشكلة التوافقية ، خوارزمية الألعاب النارية

Table des matières :

Introduction générale 1

Chapitre 1 : Recherche d'Information

1. Introduction …….. 3
2. Définition d'un Système de Recherche d'Information ….…….. 3
3. Processus de RI ……. 3
   3.1 L'indexation …….. 4
      3.1.1 Les approches d'indexation …….. 4
      3.1.2 Les étapes de l'indexation automatique …… 5
      3.1.3 Le résultat de l'indexation ……. 8
   3.2 L'appariement Document/ Requête …….. 8
   3.3 La reformulation de la requête ……. 8
      3.3.1 La reformulation manuelle …… 8
      3.3.2 La reformulation semi-automatique ……. 8
      3.3.3 La reformulation automatique ……. 8
4. Modèles de RI ……. 9
   4.1 Le modèle booléen …… 9
   4.2 Le modèle vectoriel ……..…… 9
   4.3 Le modèle probabiliste …… 10
5. Evaluation d'un système …… 11

Conclusion …….. 11

Chapitre 2 : L'optimisation combinatoire

1. Introduction ……. 14
2.Définition …… 14
3. Classification des méthodes de résolutio …… 14
   3.1 Méthodes exactes …….. 15
      3.1.1 La méthode de Branch and Bound …… 16
      3.1.2 Programmation dynamique …… 17
      3.1.3 Programmation linéaire …… 17
   3.2. Méthodes approchées …… 18
      3.2.1 Les méthodes heuristiques …….. 19
      3.2.2 Métaheuristique …….. 19
         3.2.2.1 Métaheuristiques à solution unique …… 19
         3.2.2.2 Métaheuristiques à population de solutions ……. 21

Conclusion …… 26

Chapitre 3 : Algorithme des feux d'artifice

1. Introduction …….…….. 28
2. Algorithme des feux d'artifice ……..…… 28
3. Métaphore d'inspiration …… 28
4. Motivation de l'algorithme des feux d'artifice …… 28
5. Framework général de l'algorithme des feux d'artifices …….. 29
6. Définition d'un feu d'artifice …….. 30
   6.1 Un bon feu d'artifice (Good Firework) …… 30
   6.2 Un mauvais feu d'artifice (Bad Firework) …… 30
7. Les principaux processus dans l'algorithme des feux d'artifice ……. 30

7.1 Le processus d'exploration ……..…..… 30
7.2 Le processus d'exploitation ……..…… 31
7.3 Le processus de mutation …… 31
8. Les paramètres principaux de l'algorithme des feux d'artifice ……. 31
8.1 L'opérateur d'explosion ……..…… 32
8.2 La force de l'explosion (le nombre des étincelles ……..…… 31
8.3 L'amplitude de l'explosion . …… 32
8.4 Génération des étincelles ……..…… 32
8.5 Explosion gaussienne (mutation gaussienne) ……. 33
8.6 L'opération de déplacement …… 33
9. Les variantes de l'algorithme des feux d'artifice …….. 34
Conclusion ……... 34

Chapitre 4 : l'implémentation

1. Introduction …… 36
2. La modélisation de l'approche ……. 36
3. La modélisation de l'approche …….. 36
4. La modélisation de l'approche …….. 36
5. Les packages utilises …… 36
6. Le processus d'évaluation ……. 37
7. Collection de tests ……. 38
8. Collection de tests ……. 39
9. Rappel et Précision …… 39
Conclusion ……... 41
Conclusion générale ……. 42
Bibliographie …… 44

Liste des figures

Figure I.1 Processus en U de recherche d'information …….. 4

Figure I.2 Les étapes de l'indexation automatique……... 5

figureII.1 Taxonomie des méthodes de résolution de problème d'optimisation 15

Figure II.2 Divisé en sous-problème …… 17

Figure II. 3 Organigramme de l'algorithme du recuit ……... 20

Figure II.4 Organigramme de l'algorithme tabou …… 21

Figure II.5 Fonctionnement des algorithmes génétiques. ……... 22

Figure II.6 Codage des solutions……..... 23

Figure II.7 Sélection par tournoi…… 24

Figure II.8 Croisement à un point…… 24

Figure II.9 Croisement à deux points…… 25

Figure II.10 Une mutation……. 25

Figure III.1 Framework standard de l'algorithme des feux d'artifice…….. 29

Figure III.2 Les bonnes et les mauvaises explosions d'un feu d'artifice ……. 30

Figure VI.1 : La modélisation de l'approche …… 36

Figure VI.2 : Rappel et Précision de système …… 39

Figure VI.3 : La courbe de pour la première requête …… 41

Introduction générale :

L'optimisation combinatoire occupe une place très importante en informatique. Cette importance justifie d'une part les difficultés des problèmes d'optimisation, d'autre part de nombreuses applications pratiques peuvent être formulées sous la forme d'un problème d'optimisation combinatoire. Les problèmes d'optimisation sont utilisés pour modéliser de nombreux problèmes dans les différents secteurs de l'industrie (télécommunications, électronique, mécanique, chimie, transport,…). Parmi les méthodes de résolutions des problèmes d'optimisation combinatoire nous allons choisir dans notre mémoire une nouvelle approche appelée : algorithme de feux d'artifice. Notre travail se structure de la manière suivante : après l'introduction générale ;

Le premier chapitre traite des généralités concernant le domaine de recherche d'information. Nous présentons en détail les principales approches qui ont fait preuve de performance en recherche d'information ainsi que les modèles universellement appliqués dans la mise en œuvre de ces systèmes. A la fin du chapitre nous avons expliqué la façon d'évaluer ce genre de système.

Dans le deuxième chapitre nous présentons certaines méthodes parmi les plus représentatives, développées pour résoudre les problèmes d'optimisation combinatoire.

A travers le troisième chapitre, nous allons présenter l'algorithme des feux d'artifice comme une méthode d'optimisation par essaim

Dans le quatrième chapitre, nous discuterons de la façon de modéliser l'approche proposée en termes de modélisation de solutions, ainsi que de la modélisation de la procédure d'explosion.

Enfin, nous terminons par une conclusion générale et la liste des bibliographies.

Chapitre 1 :
Recherche d'Information

1. Introduction

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie. Elle traite l'information dans la manière de l'organiser et de la façon de la sélectionner, elle peut être définie comme une activité qui dans le but de répondre à une question vise à localiser et à traiter une ou plusieurs informations au sein d'un environnement documentaire complexe.

Le traitement de cet environnement ne peut pas être effectué manuellement et donc l'objectif de la recherche d'information est d'extraire les informations pertinentes vis-à-vis d'une requête pour un utilisateur donné à travers l'utilisation d'un ensemble de programmes informatiques appelés systèmes de recherche d'information.

Dans ce premier chapitre, nous allons définir les concepts de base de la recherche d'information et les systèmes de recherche d'information (SRI).

2. Définition d'un Système de Recherche d'Information

On définit un SRI comme étant un système permettant de retrouver les documents pertinents à une requête d'utilisateur écrite dans un langage libre, à partir d'une base de documents volumineuse.

3. Processus de RI

Le processus de recherche d'informations est fourni par SRI, ce dernier met en correspondance les représentations des informations contenues dans un fond documentaire et des besoins d'utilisateur exprimés par une requête.

Cette notion de pertinence peut être appréhendée à deux niveaux :

a. ***Niveau utilisateur*:** la pertinence correspond à la satisfaction de l'utilisateur par apport à l'ensemble des documents restitués par le SRI.

b. ***Niveau système:*** le système mesure un degré de pertinence, une valeur desimilitude entre un document et une requête.

Le but de tout SRI est de rapprocher la pertinence système de la pertinence utilisateur. Pour effectuer de façon efficace cette fonction, le SRI doit réaliser l'indexation des documents, la formulation de la requête, la comparaison requête- documents et enfin la reformulation de la requête (processus non toujours présent mais important). Nous pouvons représenter schématiquement un SRI, comme illustré par la figure I.1 , par ce qui est appelé communément le processus en U de recherche d'information.

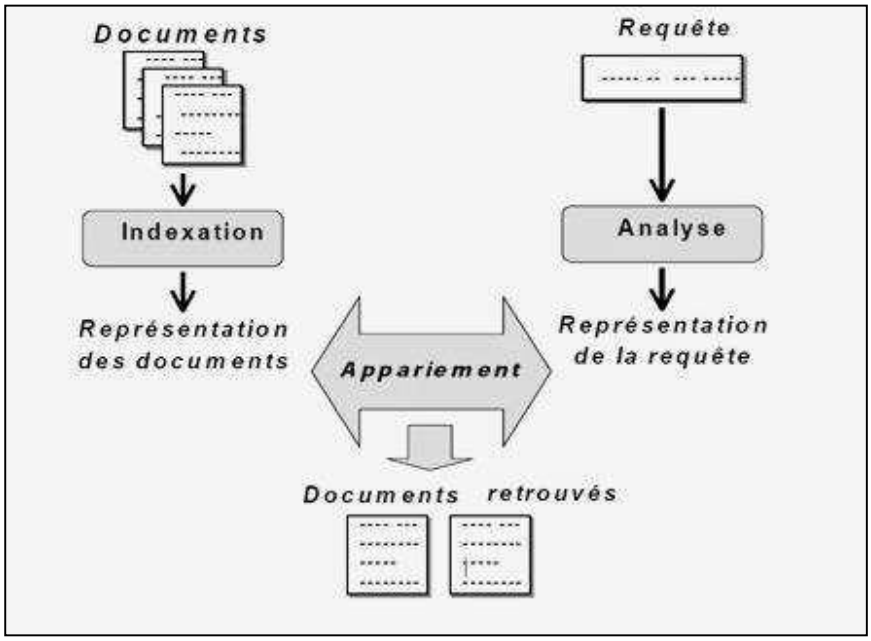


Figure I.1 Processus en U de recherche d'information

Le schéma ci-dessus montre clairement que le processus de recherche d'information se décompose en deux processus comme suit :

- ***Modèle de représentation.***
- ***Modèle de recherche d'information.***

3.1 L'indexation

Un traitement préliminaire de la base documentaire est nécessaire dans le processus de RI. Il consiste à créer un ensemble de mots clés reflétant aux mieux le contenue sémantique du document, cette liste de mots clés sera plus facilement exploitable lors du processus de la RI. Cette étape est communément appelée l'indexation.

On distingue deux types d'indexation :

–**L'indexation libre (réalisée automatiquement) :** les termes sont extraits du texteà indexer.

– **L'indexation contrôlée (réalisée manuellement ou semi-manuellement) :** qui s'appuie sur un ensemble prédéfini de termes : le thésaurus. Elle consiste à sélectionner les termes de ce thésaurus qui indexent ce texte.

3.1.1 Les approches d'indexation

✓ Indexation *manuelle :* C'est le documentaliste ou un spécialiste du domaine qui effectue l'analyse du document, pour identifier son contenu et construire une représentation de ce contenu (choix des mots effectué par des indexeurs).Elle est basée sur un vocabulaire contrôlé (lexique, liste hiérarchiques, thésaurus, ontologie).

- ✓ Indexation semi-manuelle : Le choix finale revient au spécialiste, qui intervient souvent pour choisir d'autres termes significatifs, l'indexation semi manuelle se divise en deux parties, une partie automatique permettant d'extraire une liste de descripteur, et une deuxième partie qui est manuelle réalisée par un spécialiste du domaine dont la tache est de sélectionner des termes significatifs parmi les descripteurs retournés auparavant.

- ✓ Indexation automatique : C'est le SRI qui génère les indexes des documents. L'indexation automatique a été créé afin de remédier aux problèmes liés aux approches précédentes, elle présente l'avantage d'une régularité du processus, car l'indexation automatique fournit toujours le même index pour le même document, ce qui constitue une qualité du système. En effet, l'indexation automatique pêche par son incapacité à interpréter un texte et son manque d'adaptation à de nouveaux vocabulaires. Il est impossible de trouver dans les documents autre chose que ce que le système peut détecter.

3.1.2 Les étapes de l'indexation automatique

Le processus d'indexation automatique (figure III.1) passe obligatoirement par 3 phases, chaque phase pouvant contenir une ou plusieurs étapes selon les usages des utilisateurs, c'est au programmeur de sélectionner les étapes qu'il souhaite intégrer au processus d'indexation automatique du corpus.

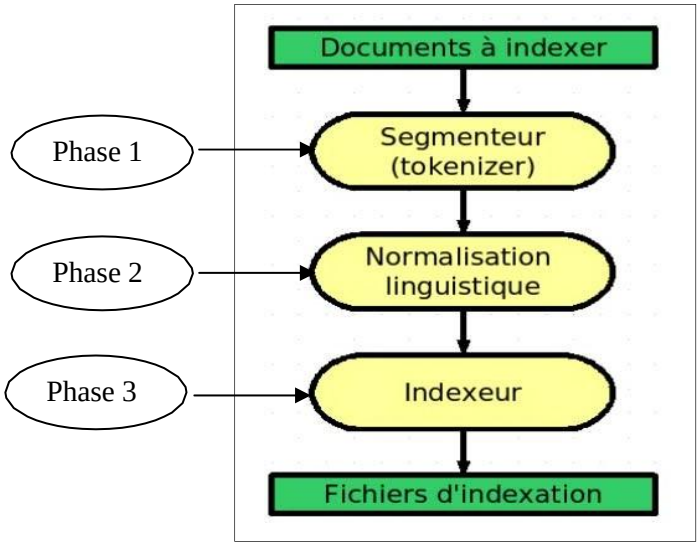


Figure I.2. Les étapes de l'indexation automatique

– ***La phase 1:*** cette phase représente la segmentation des documents en unités, cette segmentation est basée en générale sur la ponctuation et sur une liste de séparateur,le résultat de cette étape est un ensemble de mots.

– ***La phase 2 :*** cette phase peut contenir plusieurs étapes, dans ce qui suit nous

allons expliquer les étapes les plus importantes et les plus utilisées :

- Elimination des mots vides : Les mots vides sont des mots qui permettent de lier entre eux les mots d'une phrase pour la structurer (les articles, les conjonctions de coordination, les verbes auxiliaires, etc.). Ces mots ne portent pas de sens, ils ne peuvent pas constituer des index il faut donc les éliminer, cependant il ne faut pas oublier de tenir compte de certain mots vides qui auraient pour homographes des mots significatifs (ex : or).

- La racinisation (« stemming » en anglais) : consiste à rechercher la

forme tronquée d'un mot à partir de laquelle peuvent être reconstruites ses différentes variantes morphologiques. Cette opération peut être réalisée assezsimplement, en utilisant un algorithme comme l'algorithme de Porter, pour l'anglais. Par contre, elle peut entraîner une perte de sens, car la racine extraite peut être commune à des mots se rapportant à des concepts différents : Les mots port, portes (ouverture) et portera ont la même racine port mais se rapportent à trois concepts différents, la racinisation permet d'augmenter le rappel mais peut diminuer la précision.

- La lemmatisation : les mots d'une langue peuvent être classés en deux catégories : les lemmes: formes canoniques (infinitif pour les verbes, singulier pour les noms, etc.) qui constituent en général les entrées dans un dictionnaire de cette langue, les mots obtenus par flexion de ces lemmes: conjugaison d'un verbe, changement de genre ou de nombre, etc. Par exemple, le mot « devrait » estobtenu par flexion (conditionnel présent, 3e personne du singulier) du verbe « devoir ».La lemmatisation consiste à remplacer un mot par son lemme, les mots port, portes et portera seront remplacés par leurs lemmes : port, porter ou porte selon le contexte et porter. C'est une opération plus couteuse que la racinisation car elle nécessite une analyse morphologique et syntaxique des phrases.

- Extraction des mots composés : Il est important de reconnaître les mots composés car ce sont des unités de sens. Par exemple : « arbre à cames » ou « pomme de terre ».

- Extraction des entités nommées : les entités nommées sont des mots ou des groupes de mots qui désignent des personnes, des organisations, des dates, deslieux, etc. Par exemple, si un texte contient l'expression : « 5 juillet 1962 » il est plus intéressant de l'indexer globalement par cette date plutôt que les trois termes:

« 5 », « juillet » et « 1962 ».Si de plus l'indexeur est capable de reconnaître que c'est la date de l'indépendance de l'Algérie, l'indexation sera encore plus précise.

– ***L'étape3 :*** Dans cette phase on utilise une approche permettant de sélectionner les index et de leur associer une pondération, cette dernière permet d'assigner aux

termes leur degrés d'importance dans les documents, **il existe 3 approches pour le choix des index :**

- Approche basée sur la fréquence d'occurrences : Cette approche consisteà choisir les mots représentants selon leur fréquence d'occurrence. La façon laplus simple consiste à définir un seuil sur la fréquence: si la fréquence d'occurrence d'un mot dépasse ce seuil, alors il est considéré important pour le document. Cependant, en calculant ces fréquences, on s'aperçoit que les mots les plus fréquents sont des mots fonctionnels (mots vides), comme l'explique la loi deZipf.

- Approche basée sur la valeur de discrimination : Par "discrimination",on réfère au fait qu'un terme distingue bien un document des autres documents. C'est-à-dire, un terme qui a une valeur de discrimination élevée doit apparaître seulement pour un petit nombre de documents. L'idée est de garder seulement les termes discriminants, et éliminer ceux qui ne le sont pas. Le calcul de la valeur de discrimination a été développé dans le modèle vectoriel.

Pour calculer la valeur de discrimination d'un terme, on doit comparer une sorte d'uniformité au sein du corpus avec celle du corpus transformé dans lequel leterme en question a été uniformisé (mis au même poids). L'idée est que, si on uniformisant le poids d'un terme dans tous les documents, on obtient une grande amélioration dans l'uniformité du corpus, ce terme était donc très différent (non uniformément distribué) dans différents documents. Il a donc une grande valeurde discrimination. En revanche, si en uniformisant le poids du terme, on n'obtient pas beaucoup d'amélioration sur l'uniformité, ce terme était donc déjà distribué de façon uniforme, donc peu discriminant.

- Approche basée sur tf*idf : tf*idf désigne un ensemble de schémas de pondération (et de sélection) de termes. tf signifie "term frequency" et idf "inverted document frequency". Par tf, on désigne une mesure qui a rapport à l'importance d'un terme pour un document. En général, cette valeur est déterminéepar la fréquence du terme dans le document. Par idf, on mesure si le terme est discriminant (ou non-uniformément distribué). Ici, on donne quelques formules detf et d'idf souvent utilisées.

- tf = fréquence d'occurrence du terme dans un document f(t, d);
  - tf = f(t, d) / Max [f(t, d)] où Max [f(t, d)]
  - tf = log (f(t, d))
  - tf = log (f(t, d) + 1)

idf = log (N/n) où N est le nombre de documents dans le corpus, et n ceux qui contient le terme

Une formule tf*idf combine les deux critères qu'on a vus: l'importance du terme pour un document (par tf), et le pouvoir de discrimination de ce terme (par idf). Ainsi, un terme qui a une valeur de tf*idf élevée doit être à la fois important dans ce document, et aussi il doit apparaître peu dans les autres documents.

3.1.3 Le résultat de l'indexation

Le résultat d'une indexation est donc un ensemble de termes qui peuvent être soit un mot, soit une racine de mot (soit un terme composé si on possède un mécanisme pour reconnaître des termes composés).

3.2 L'appariement Document/ Requête

La correspondance entre les termes de la requête d'un utilisateur et ceux des documents s'effectue au niveau de l'appariement document-requête, cette étape sert à renvoyer une liste de documents ordonnées selon un degré de pertinence, ce dernier est calculé à partir d'une fonction notée RSV (Q, D) (Retrieval Status Value), ou Q est une requête et D un document. L'expression de la fonction d'appariement est tributaire du modèle de RI choisi.

3.3 La reformulation de la requête

Le but essentiel d'un système de recherche d'information est de permettre à l'utilisateur d'avoir un résultat satisfaisant (des documents pertinents) par apport à son besoin exprimé par une requête. La possibilité de reformuler la requête initiale s'avère intéressante dans le processus de la RI. Cela fera en sorte que le résultat retourné soit plus pertinent. Il existe trois méthodes de reformulation de requêtes :

3.3.1 La reformulation manuelle

Elle consiste à présenter à l'utilisateur une liste de documents jugés pertinents en réponse à la requête initiale. C'est à l'utilisateur de sélectionner à partir des documents pertinents ceux dont lesquels le système va extraire les termes à rajouter à la requête initiale dans le but d'effectuer une nouvelle recherche.

3.3.2 La reformulation semi-automatique

Cette technique nécessite l'intervention de t'utilisateur qui doit identifier et sélectionner les documents pertinents et les documents non pertinents

3.3.3 La reformulation automatique

L'extension de la requête est faite sans intervention de l'utilisateur grâce à l'utilisation d'un thésaurus contenant des informations de type linguistique (équivalence, association, hiérarchie) et statistique (pondération des termes).

4. Modèles de RI

Si c'est l'indexation qui choisit les termes pour représenter le contenu d'un document ou d'une requête, c'est au modèle de leur donner une interprétation. Étant donné un ensemble de termes pondérés issus de l'indexation, le modèle remplit les deux rôles suivants:

- Créer une représentation interne pour un document ou pour une requête baséesur ces termes.
- Définir une méthode de comparaison entre une représentation de document et une représentation de requête afin de déterminer leur degré de correspondance.

Le modèle joue un rôle central dans la RI. C'est le modèle qui détermine le comportement clé d'un système de RI.

4.1 Le modèle booléen

Le modèle booléen est basé sur la théorie des ensembles. Dans ce modèle, les documentset les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document. Un exemple de représentation d'un document est comme suit : d = t1 At2At3...A tn.

Une requête est une expression booléenne dont les termes sont reliés par des opérateurs logiques (OR, AND, NOT) permettant d'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 comme suit : " q = (t1At2) v (t3At4) " . La fonction de correspondance est basée sur l'hypothèse de présence/absence des termes de la requête dansle document et vérifie si l'index de chaque document dj 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}.

4.2 Le modèle vectoriel

Dans ces modèles, 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 des vecteurs d'un espace à *n* dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation.

L'index d'un document *dj* est le vecteur = (w1,j ,w2,j ,w3,j , ..,wn,j), où wk,j ∈ [0 1]

dénote le poids du terme *tk* dans le document *dj* . Une requête est également représentée par un vecteur = (w1, q, w2, q, w3, q,.., wn, q), où *wk, q* est le poids du terme tk dans la requête q.

La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents. 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.

Plus les vecteurs sont similaires, plus l'angle formé est petit, et plus le cosinus de cet angle est grand. A l'inverse du modèle booléen, la fonction de correspondance évalue une correspondance partielle entre un document et une requête, ce qui permet de retrouver des documents qui ne reflètent pas la requête qu'approximativement. Les résultats peuvent donc être ordonnés par ordre de pertinence décroissante.

4.3 Le modèle probabiliste

Ce modèle est fondé sur le calcul de la probabilité de pertinence d'un document pour une requête . Le principe de base consiste à retrouver des documents qui ont en même temps une forte probabilité d'être pertinents, et une faible probabilité d'être non pertinents. Etant donné une requête utilisateur *Q* et un document *D*, il s'agit de calculer la probabilité de pertinence du document pour cette requête.

Deux possibilités se présentent : *R, D* est pertinent pour *q* et D, n'est pas pertinent pour q. Les documents et les requêtes sont représentés par des vecteurs booléens dans un espace à n dimensions. Un exemple de représentation d'un document dj et une requête q est le suivant :

dj = (w1, j, w2, j, w3, j, .., wn, j),

q = (w1, q, w2, q, w3, q, .., wn, q). Avec wk, j ∈ [0 1] et wk, q ∈ [0 1].

La valeur de wk, j (resp. wk, q) indique si le terme t_k apparaît ou non dans le document *dj*

(resp. q).

Le modèle probabiliste évalue la pertinence du document dj 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 soit non pertinent pour *q*, notée *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 de probabilités conditionnelles selon qu'un terme de la requête est présent, dans un document pertinent ou dans un document non pertinent. Cette mesure de similarité entre la requête et les documents peut se calculer 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, leur rareté et la longueur des documents

5. Evaluation d'un système

La qualité d'un système doit être mesurée en comparant les réponses du système avec les réponses idéales que l'utilisateur espère recevoir. Plus les réponses du système correspondent à celles que l'utilisateur espère, mieux est le système.

L'évaluation constitue donc une étape importante lors de la mise en œuvre d'un modèle de recherche d'information puisqu'elle permet de paramétrer le modèle.

Conclusion

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'information, des systèmes de recherche d'information et ceux des outils de recherche sur le web.

A travers les différentes sections que nous avons présentées, nous concluons que la recherche d'information, s'attache à définir des modèles et des systèmes afin de faciliter l'accès à un ensemble de documents se trouvant dans des bases documentaires ou encore sur le web. Le but est de permettre aux utilisateurs de retrouver les documents dont le contenu répond à leur besoin en information, il s'agit donc de retourner l'ensemble de documents pertinents. Cependant, nous constatons que la notion de pertinence dépend de la satisfaction de l'utilisateur d'une part, et des différents sens portés par les termes de la requête d'une autre part. Cette constatation constitue le point faible de la recherche

d'information classique, elle représente également le point de départ pour de nouveaux paradigmes de recherche qu'on vas l'étudié dans le deuxième chapitre qui est l'optimisation combinatoire.

Chapitre 2 :
L'optimisation combinatoire

1. Introduction :

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. Etant donné l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) et les méthodes approchées (incomplètes)

Dans ce chapitre nous présentons certaines méthodes parmi les plus représentatives, développées pour résoudre les problèmes d'optimisation combinatoire.

2. Définition

Un problème d'optimisation combinatoire consiste à trouver la *meilleure* solution dans un ensemble discret dit ensemble des solutions réalisables. En général, cet ensemble est fini mais compte un très grand nombre d'éléments, et il est décrit de manière implicite, c'est-à-dire par une liste, relativement courte, de contraintes que doivent satisfaire les solutions réalisables.

Pour définir la notion de *meilleure solution*, une fonction, dite *fonction objectif*, est introduite. Pour chaque solution, elle renvoie un réel et la meilleure solution (ou *solution optimale*) est celle qui minimise ou maximise la fonction objectif. Clairement, un problème d'optimisation combinatoire peut avoir plusieurs solutions optimales.

Exemple : Le problème du plus court chemin entre deux sommets A et B d'un graphe est un exemple classique de problème d'optimisation combinatoire. L'ensemble des solutions réalisables est l'ensemble des chemins entre A et B tandis que la fonction objectif est la longueur du chemin.

3. Classification des méthodes de résolution

Les méthodes de l'optimisation combinatoire peuvent être classées en deux grandes familles de classes : les méthodes exactes et les méthodes approchées. La figure II.1 illustre la taxonomie des méthodes de résolution des problèmes d'optimisation.

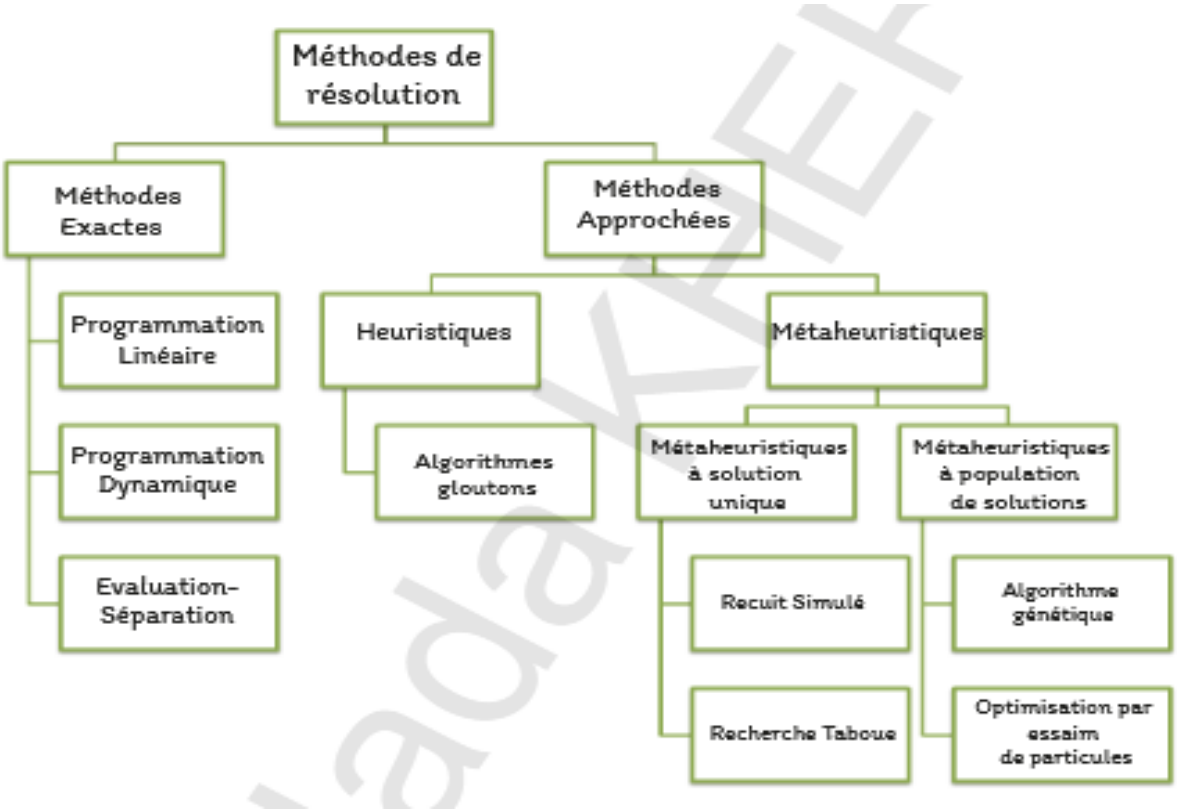


figure II.1 Taxonomie des méthodes de résolution de problème d'optimisation

3.1 Méthodes exactes

Les méthodes exactes (appelées aussi complètes) produisent une solution optimale pour une instance de problème d'optimisation donné. Elles se reposent généralement sur la recherche arborescente et sur l'énumération partielle de l'espace de solutions.

Elles sont utilisées pour trouver au moins une solution optimale d'un problème. Les algorithmes exacts les plus connus sont la méthode de séparation et évaluation, la programmation dynamique et la programmation linéaire.

L'inconvénient majeur de ses méthodes est l'explosion combinatoire : Le nombre de combinaisons augmente avec l'augmentation de la dimension du problème. L'efficacité de ces algorithmes n'est prometteuse que pour les instances de

problèmes de petites tailles.

Parmis ces algorithmes, on cite :

- ✓ Les méthodes de recherche arborescente ou évaluation séparé (Branch and Bound),
- ✓ La programmation dynamique,
- ✓ La programmation linéaire.

3.1.1 La méthode de Branch and Bound

La méthode de Branch and Bound (procédure par évaluation et séparation progressive) consiste à énumérer des solutions, en utilisant certaines propriétés du problème en question, cette technique arrive à éliminer des solutions partielles qui ne mènent pas à la solution que l'on recherche. De ce fait, on arrive souvent à obtenir la solution recherchée en des temps raisonnables.

Pour ce faire, cette méthode se dote d'une fonction qui permet de mettre une borne sur certaines solutions pour, soit les exclure, soit les maintenir comme des solutions potentielles. Bien entendu, la performance d'une méthode de Branch and Bound dé- pend, entre autres, de la qualité de cette fonction.[6]

- ➢ **Algorithme général :**

Début

Placer le nœud début de longueur *0* dans une liste.

Répéter

Si la première branche contient le nœud recherché **alors**

Fin avec succès.

Sinon

- Supprimer la branche de la liste et former des branches nouvelles en étendant la branche supprimée d'une étape.
- Calculer les coûts cumulés des branches et les ajouter dans la liste de telle sorte que la liste soit triée en ordre croissant.

Jusqu'à (liste vide ou nœud recherché trouvé)

Fin

3.1.2 Programmation dynamique

La programmation dynamique a été appelée comme cela depuis 1940 par Richard Bellman et permet d'appréhender un problème de façon différente de celle que l'on pourrait imaginer au premier abord. Le concept de base est simple : une solution optimale est la somme de sous-problèmes résolus de façon optimale. Il faut donc diviser un problème donné en sous-problèmes et les résoudre un par un.[8]

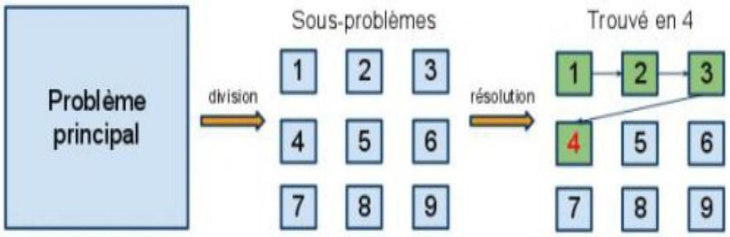


Figure II.2 – Divisé en sous-problème

- **Algorithme général de programmation dynamique :**

La conception d'un algorithme de programmation dynamique peut être planifiée dans uneséquence de quatre étapes.

1. Caractériser la structure d'une solution optimale.
2. Définir récursivement la valeur d'une solution optimale.
3. Calculer la valeur d'une solution optimale en remontant progressivement jusqu'à l'énoncé du problème initial.
4. Construire une solution optimale pour les informations calculées.

3.1.3 Programmation linéaire

En mathématique les problèmes de programmation linéaire (PL) sont des problèmes d'optimisation où la fonction objective et les contraintes sont toutes linéaires. La programmation linéaire désigne également la manière de résoudre les problèmes linéaires. La programmation linéaire est un domaine central de l'optimisation car les problèmes de PL sont des problèmes d'optimisation les plus faciles toutes les contraintes y étant linéaires beaucoup de problèmes réel de recherche opérationnelle

peuvent être exprimés comme problèmes de PL.

Pour cette raison un grand nombre d'algorithmes pour la résolution d'autres problèmes d'optimisation sont fondés sur la résolution de problèmes linéaires. Le terme programmation linéaire suppose que les solutions à trouver doivent être représentées en variable réelle. S'il est nécessaire d'utiliser des variables discrètes dans la modélisation du problème (contraintes dites d'intégrité), on parle alors de programmation linéaire en nombres entiers. Il est important de savoir que ces derniers sont nettement plus complexes à résoudre que les (PL) à variables contin

3.2. Méthodes approchées

Contrairement aux méthodes exactes, les méthodes approchées ne fournissent pas forcément une solution optimale, mais seulement une bonne solution (de qualité raisonnable) dans un temps raisonnable.

Les méthodes approchées englobent deux classes :

- Les méthodes constructives ou heuristiques.
- Les métaheuristiques

3.2.1 Les méthodes heuristiques

Définition d'une heuristique

Le mot heuristique, dérivé de la langue grec, vient du verbe heuriskein qui signifie trouver. Les méthodes itératives sont celles qui construisent une solution pièce par pièce.A partir d' une partielle solution qui est d' abord vide, ils tentent d' étendre la partie solution de la précédente étape à chaque étape, et ce processus est répété jusqu'à ce que une complète solution est obtenue. Les méthodes constructives sont généralement utilisables quand la qualité de solution n'est pas un facteur primordial ou la taille de l'instance est raisonnable, en l'occurrence pour générer une solution initiale dans une métaheuristique, ces méthodes rapides et faciles d'implémentation.

L'Algorithme glouton :

La construction d' une pratique solution basée sur une série de décisions qui sont prises mieux chaque fois en fonction sur un locale critère, sans appel en question précédentes

décisions. Dans la plupart des cas, la solution obtenue est acceptée. L'avantage de cette methode est : algorithmes simples à implémenter, et l'Inconvénient : solutions approchées obtenues plus ou moins bonnes, critère local .

3.2.2 Métaheuristique

Le mot métaheuristique est composé de deux mots grecs : méta et heuristique. Le mot méta est un suffixe signifiant *au-delà* c'est-à-dire *de niveau supérieur*.

Les métaheuristiques sont des méthodes généralement inspirées de la nature. Contrairement aux heuristiques, elles s'appliquent à plusieurs problèmes de nature différente. Pourcela on peut dire qu'elles sont des heuristiques modernes, de plus haut niveau, dédiées particulièrement à la résolution des problèmes d'optimisation. Leur but est d'atteindreun optimum global tout en échappant les optima locaux.

Les métaheuristiques regroupent des méthodes qui peuvent se diviser en deux classes:

3.2.2.1 Métaheuristiques à solution unique : Ces méthodes traitent une seule solutionà la fois, afin de trouver la solution optimale.

a) Le recuit simulé :

La méthode du circuit simulé est une généralisation de la méthode Monte-Carlo , dans le but de trouver la meilleure solution à un problème donné .Elle a été développée par trois IBM chercheurs, S. Kirkpatrick, CD Gelatt, et MP Vecchi, en 1983, et indépendamment par V. Cerny en 1985, basée sur le Metropolis algorithme, qui permet un de décrire l' évolution d' un thermodynamique système. [5].

La principale idée derrière la simulation recuit, comme proposé par Metropolis en 1953, est de simuler le comportement de la matière dans un recuit processus qui est largement utilisé dans la métallurgie. Le but est d'atteindre un équilibre thermodynamique; cet équilibre (où l' énergie est minimale) représente la solution optimale à un problème dans la méthode de simulation . Le système de l' énergie sera être calculé en utilisant un coût fonction (ou fonction Objectif). Kirkpatrick, En gros le principe consiste à générer successivement des configurations à partir d'une solution initiale S_0 et d'une température initiale T_0 qui diminueratout au long du processus jusqu'à atteindre une température finale ou un état d'équilibre (optimum global).

- **Algorithme général de recuit simulé :**

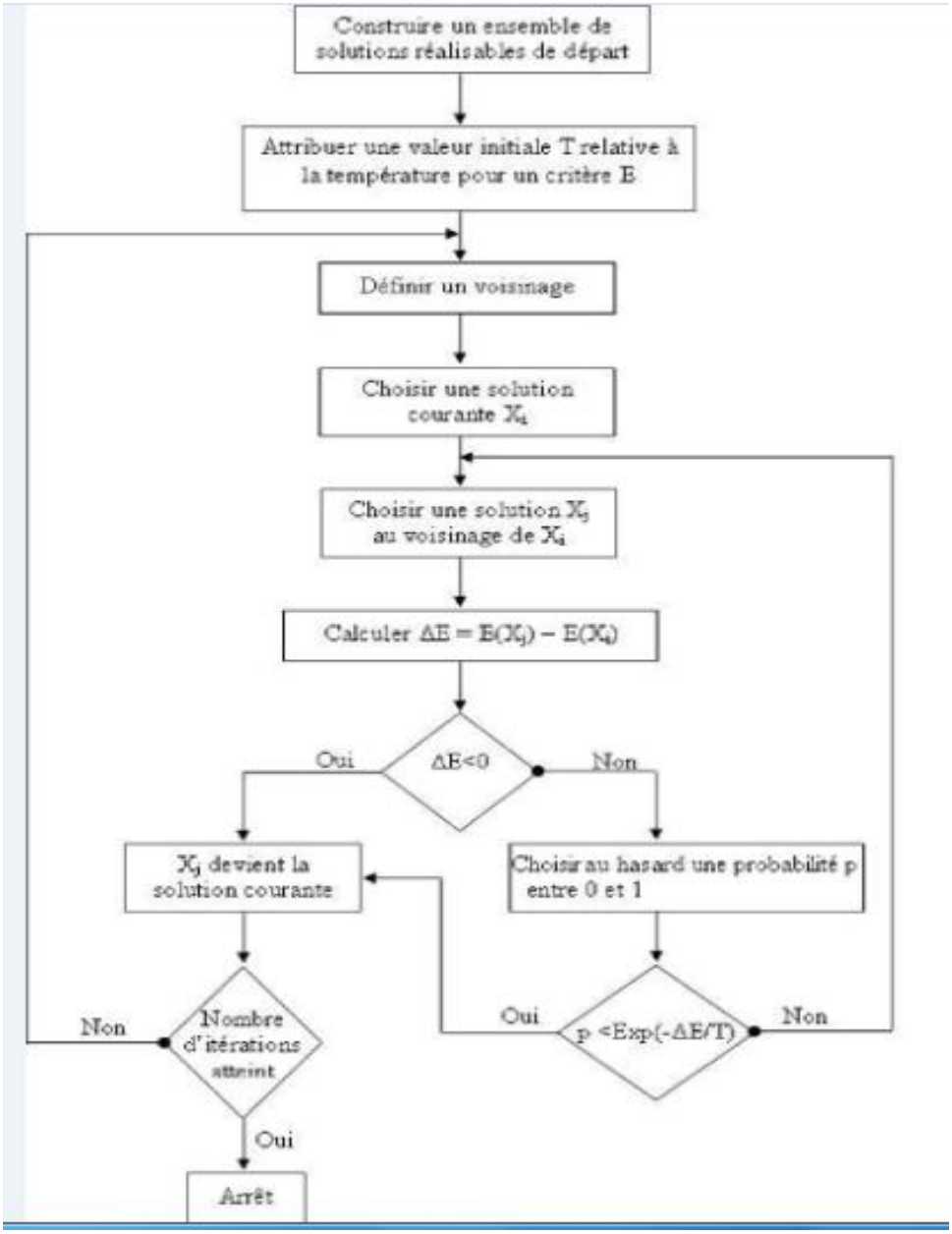


Figure II. 3: Organigramme de l'algorithme du recuit

b) La recherche tabou :

F. Glover [6] a formalisé la méthode tabou de recherche, ou simplement méthode tabou .Sa principale caractéristique distinctive est la mise en œuvre de mécanismes inspirés de la mémoire humaine .L' idée est de garder trace de votre voyage dans une mémoire et reportez - vous à ce quand vous cherchez pour quelque chose**.**

➢ **Notion de base :**

À chaque itération, l'algorithme tabou choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction de coût. Pour cette raison, on dit de la recherche avec tabou

qu'elle est une méthode agressive.

➢ **La liste tabou (mémoire) :**

L'idée de base derrière la liste taboue est de mémoriser les configurations ou régions qui ont été visitées et de mettre en place des mécanismes qui empêchent la recherche de revenir trop rapidement à ces configurations.

➢ **Algorithme générale :**

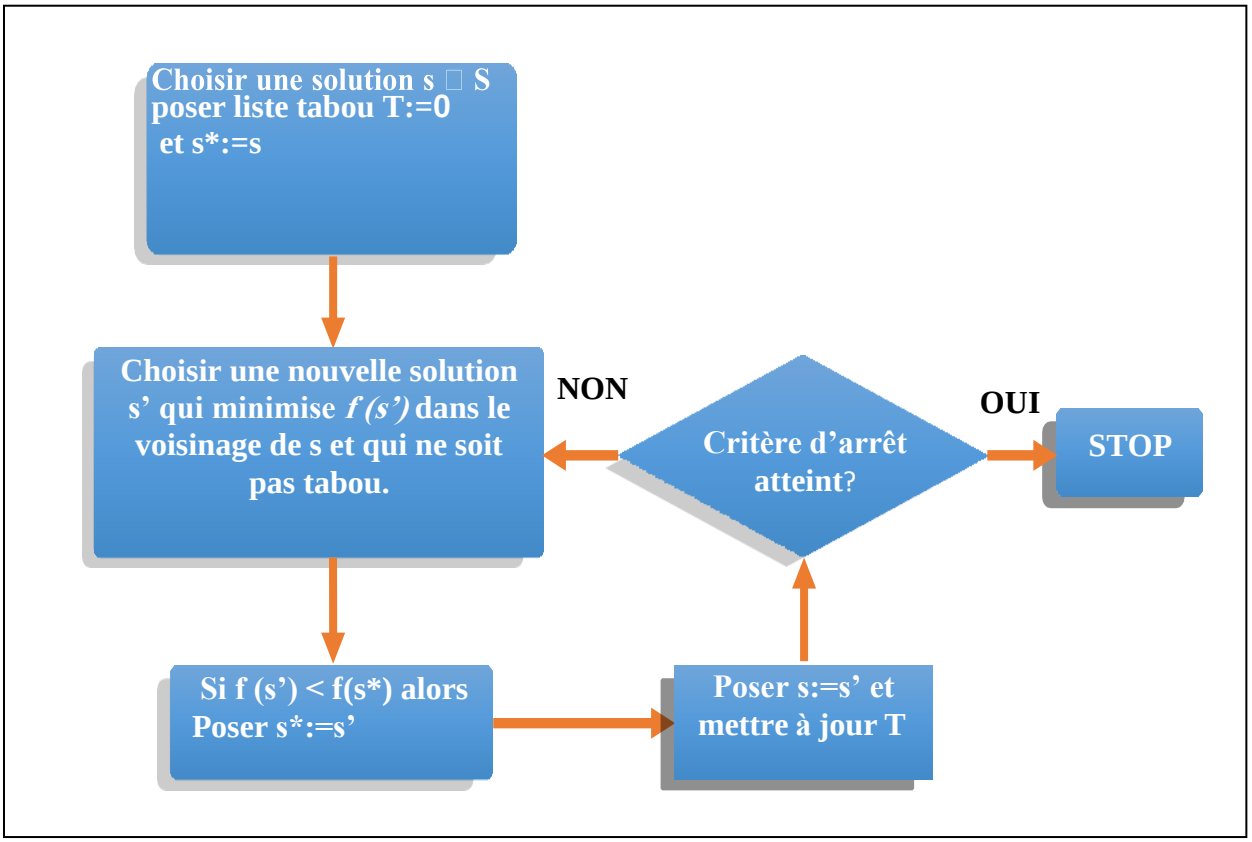


Figure II.4: Organigramme de l'algorithme tabou

3.2.2.2 Métaheuristiques à population de solutions : Ces méthodes utilisent une population de solutions à chaque itération jusqu'à l'obtention de la solution globale.

b) Les algorithmes génétiques :

Les algorithmes génétiques sont inspirés de la théorie de l'évolution et des processus biologiques qui permettent à des organismes de s'adapter à leur environnement.

Ils ont été inventés dans le milieu des années 60 (Holland, 1962; Rechenberg, 1965; Fogel et al, 1966) [8].

La sélection naturelle, que Darwin appelle l'élément "propulseur" de l'évolution, favorise les individus d'une population qui sont le mieux adaptés à un environnement. La sélection est suivie de croisements et de mutations au niveau des individus, constitué d'un ensemble de gènes. Ainsi deux individus "parents", qui se croisent, transmettent une partie de leur patrimoine génétique à leurs descendants. L'individu enfant fait que celui-ci est plus au. Au fur et à mesure des générations son sélectionne les individus les mieux adaptés, etl'augmentation du nombre des individus bien adaptés fait évoluer la population entière [9].

La mise en œuvre des algorithmes génétiques nécessite plusieurs étapes à détailler. La première est le codage d'un individu représenté par un chromosome. La seconde est le calcul de la qualité. La troisième est de définir les opérateurs de reproduction

➢ **Algorithme générale :**

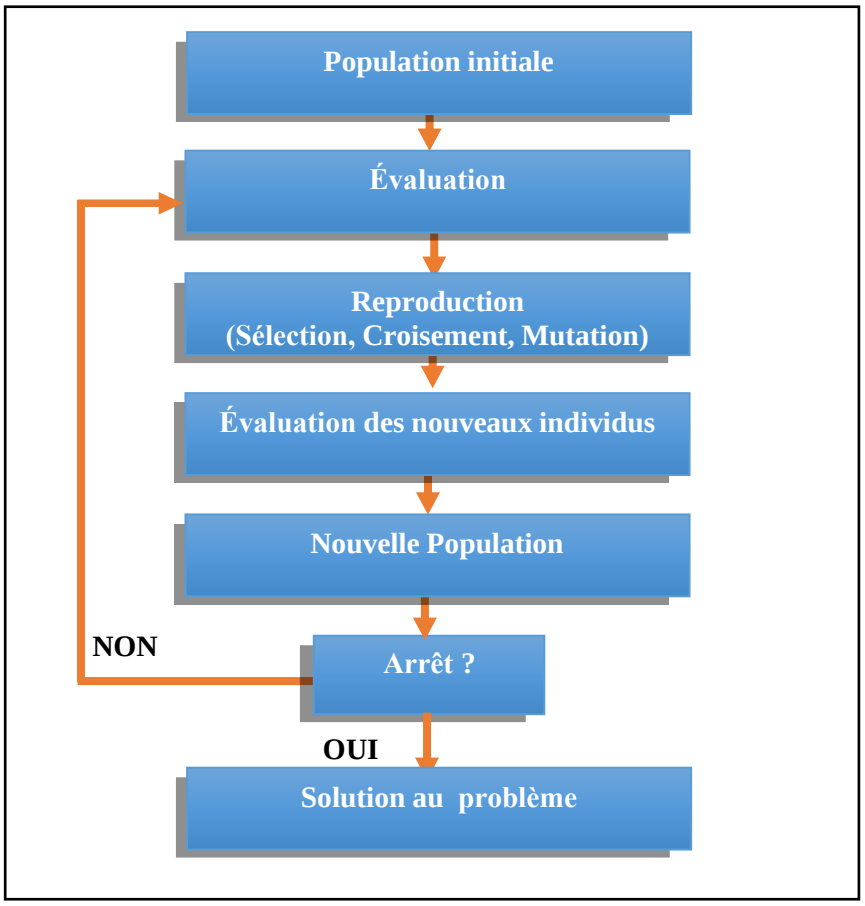


Figure II.5: Fonctionnement des algorithmes génétiques.

b) Codage des individus :

Un principe de codage de l'élément de population. Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques. Les codages binaires ont été très utilisés à l'origine. Les codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs pour l'optimisation de problèmes à variables réelles.

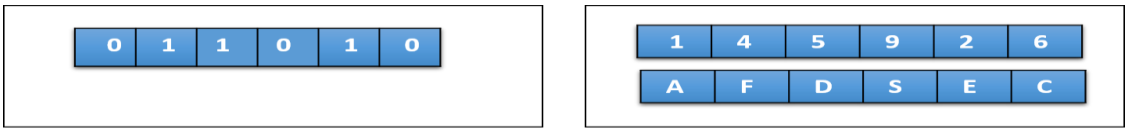


Figure II.6: Codage des solutions.

Les algorithmes génétiques utilisent trois opérateurs pour générer de nouvelles solutions :

1. L'opérateur de sélection qui permet de choisir des solutions parentes sur lesquelles lareproduction va être faite pour générer des nouvelles solutions.
2. L'opérateur de croisement qui permet de croiser les deux solutions parentes et créer denouvelles solutions.
3. L'opérateur de mutation qui permet de diversifier les nouvelles solutions afin qu'elles neressemblent pas trop aux solutions parentes.

L'opérateur de sélection :

La sélection consiste à choisir des individus qui permettront de générer de nouveaux individus. Plusieurs méthodes existent pour sélectionner des individus destinés à la reproduction. On citera les deux méthodes classiques les plus utilisées.

La sélection par la roulette :

La sélection des individus par le système de la roulette s'inspire des roues de loterie [10]. À chacun des individus de la population est associé un secteur d'une roue. L'angle du secteur étant proportionnel à la qualité de l'individu qu'il représente. Vous tournez la roue et vous obtenez un individu. Les tirages des individus sont ainsi pondérés par leur qualité. Et presque logiquement, les meilleurs individus ont plus de chance d'être croisés et de participer à l'amélioration de notre population. illustreune population de 5 individus dont les performances sont représentées en roulette.

La sélection par tournoi : Le principe de la sélection par tournoi augmente les chances pour les individus de piètre qualité de participer à l'amélioration de la population. Le principe est très rapide à implémenter. Un tournoi consiste en une rencontre entre plusieurs individus pris au hasard dans la population. Le vainqueur du tournoi est l'individu de meilleure qualité. Cette méthode est en général satisfaisante .

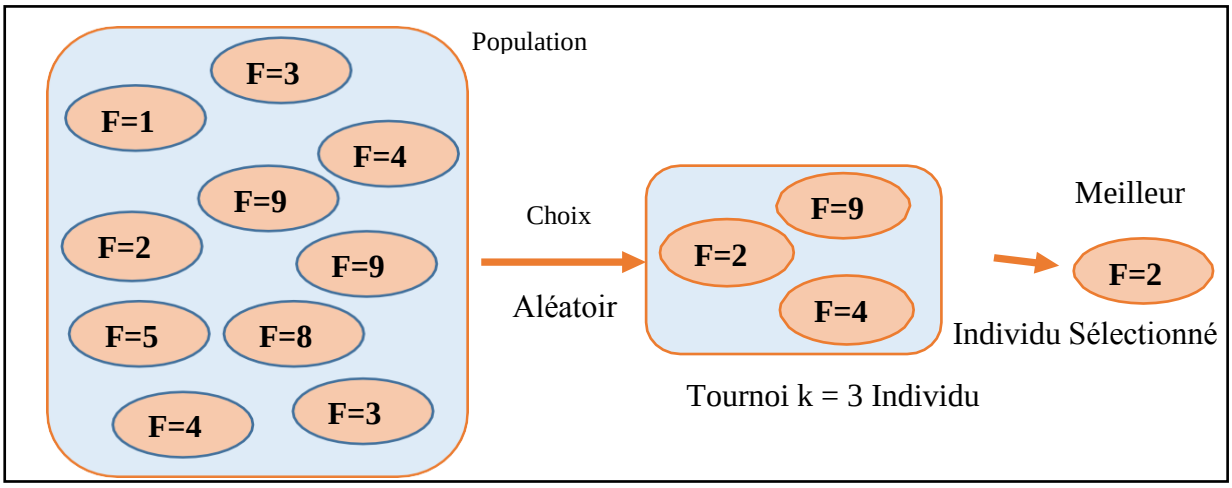


Figure II.7 Sélection par tournoi.

L'opérateur de croisement:

Les croisements permettent de simuler des reproductions d'individus dans le but d'encréer des nouveaux. Il est tout à fait possible de faire des croisements aléatoires. Toutefois, une solution largement utilisée est d'effectuer des croisements multi-points.

Le croisement à un point : Il a été initialement défini pour le codage binaire. Le principe consiste à tirer aléatoire une position pour chaque parent et à échanger les sous- chaines des parents à partir des positions tirées. Ce qui donne naissance à deux nouveaux individus ind 1 et ind 2 (Figure 8).

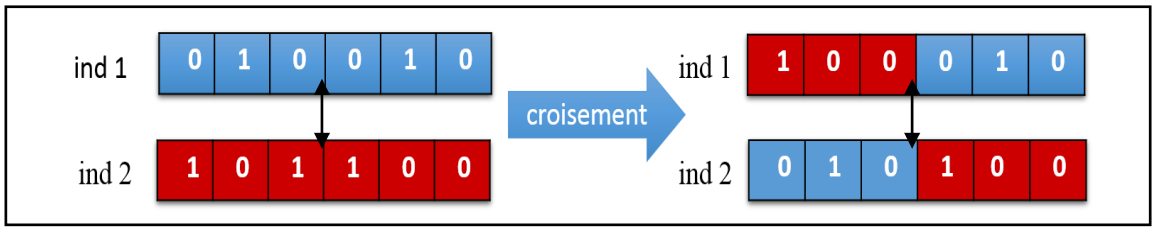


Figure II.8 Croisement à un point.

Les croisements multi-points : Elle reprend le mécanisme de la méthode de croisement à

unpoint en généralisant l'échange à 3 ou 4 sous chaines.

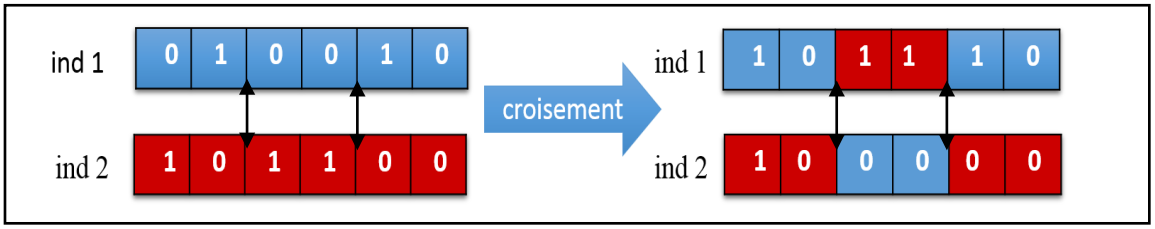


Figure II.9 : Croisement à deux points.

L'opérateur de Mutation :

L'opération de mutation protège les algorithmes génétiques des pertes prématurées d'informations pertinentes. Elle permet d'introduire une certaine information dans la population, qui a pu être perdue lors de l'opération de croisement. Ainsi elle participe au maintien de la diversité, utile à une bonne exploration du domaine de recherche (Figure 10).

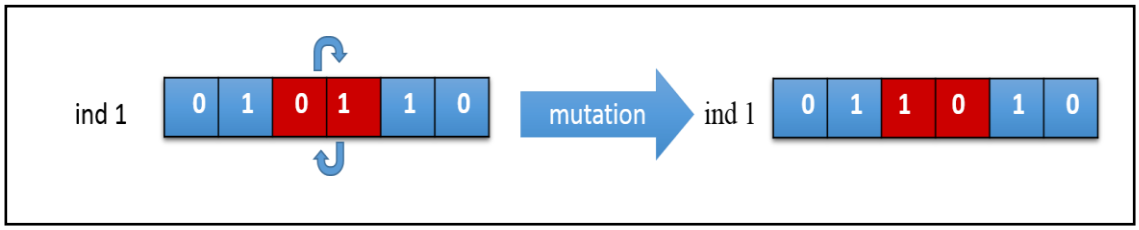


Figure II.10 Une mutation.

L'opérateur de remplacement :

Cet opérateur est consisté à réintroduire les descendants obtenus par application successive des opérateurs de sélection, de croisement et de mutation (la population P') dans la population de leurs parents (la population P). On trouve des méthodes de remplacement différentes par exemple méthode de remplacement stationnaire **[11]**.

Le remplacement stationnaire : dans ce cas, les enfants remplacent automatiquement les parents sans tenir compte de leurs performances respectives, et le nombre d'individus de la population ne varie pas tout au long du cycle d'évolution simulé, ce qui implique donc d'initialiser la population initiale avec un nombre suffisant d'individus. Cette méthode peutêtre mise en œuvre de 2 façons différentes :

1. La première se contente de remplacer la totalité de la population P par la population P',

cette méthode est connue sous le nom de remplacement générationnel**.**

2. La deuxième méthode consiste on remplace les mauvaise individus trouvées dans la population P par les meilleurs de la population P'.

Conclusion

Nous avons présenté au cours de ce chapitre l'optimisation combinatoire de façon détaillé ; nous parlons sur la notion du problème d'optimisation combinatoire et qu'ils peuvent être réparties en deux grandes classes de méthodes pour la résolution des problèmes : Les méthodes exactes, Les méthodes approchées et chacun de ces méthodes diviser en plusieurs classes. Dans le chapitre qui suit, nous allons présenter l'algorithme des feux d'artifice comme une méthode d'optimisation combinatoire.

Chapitre 3 : Algorithme des feux d'artifice

1. Introduction :

L'optimisation Combinatoires joue un rôle plus important dans la recherche opérationnelles et l'informatique scientifique en général. Plusieurs problèmes, en fait, peuvent être classés comme des problèmes d' optimisation combinatoire .L'utilisation des méthodes exactes pour la résolution de ces problèmes est très coûteuse en temps, voir dans certain cas impossible dû à leur temps d'exécution exponentiel. De ce fait les recherches se sont orientées vers l'utilisation des métaheuristiques permettant des gains en temps et en espace. A travers de ce chapitre, nous allons présenter l'algorithme des feux d'artifice comme une méthode d'optimisation par essaim.

2. Algorithme des feux d'artifice

L'algorithme des feux d'artifice forme une nouvelle classe des métaheuristiques récemment proposée pour résoudre les problèmes d'optimisation difficile. Cet algorithme s'inspire par l'observation des comportements des explosions des feux d'artifice dans le ciel à la nuit. Souvent, les algorithmes des feux d'artifice sont classés comme étant une nouvelle branche de l'intelligence en essaime (Swarn Intelligence).

Dans ce qui suit, nous allons introduire cette nouvelle méthode d'optimisation, sa métaphore d'inspiration, ses principales caractéristiques, et nous présenterons ses principaux domaines d'application.

3. Métaphore d'inspiration:

Le premier algorithme de ce type a été proposé par Ying Tan et Yuanchun Zhu en 2010 [3], à travers l'observation du fait que l'explosion d'un feu d'artifice est similaire à la façon dont un individu cherche une solution optimale dans un algorithme issue de l'intelligence en essaim.

4. Motivation de l'algorithme des feux d'artifice

L'algorithme des feux d'artifice est un nouvel algorithme récemment développé de nature de l'intelligence en essaim. Il prend sa base théorique par la simulation du processus d'explosion des feux d'artifice. En analogie avec de véritables feux d'artifice explosant et illuminant le ciel nocturne, les feux d'artifice (c'est-à-dire les individus) dans cet algorithme sont laissés à l'espace de recherche potentiel. Pour chaque feu d'artifice, un processus d'explosion est initié et une pluie d'étincelles remplit l'espace local qui l'entoure. Les feux d'artifice ainsi que les nouvelles étincelles générées représentent des solutions

potentielles dans l'espace de recherche.

Par analogie à d'autres algorithmes d'optimisation, l'objectif est de trouver une bonne solution (idéalement l'optimale) d'un problème d'optimisation combinatoire, qui est généralement définie par une fonction objective.

5. Framework général de l'algorithme des feux d'artifices

Quand un feu d'artifice est déclenché, une pluie d'étincelles remplira l'espace local autour du feu d'artifice. Selon les auteurs de cet algorithme le processus d'explosion d'un feu d'artifice est considéré comme étant une recherche locale dans l'espace autours d'un point bien déterminé où le feu d'artifice est déclenché par les étincelles générées dans l'explosio.

La figure ci-dessous présente le framework standard de l'algorithme des feux d'artifice.

Au début, N feux d'artifice (Fireworks) sont initialisés aléatoirement, et leur qualité (c.-à-d. Leur condition physique) est évaluée afin de déterminer l'amplitude de l'explosion et le nombre d'étincelles pour chaque feu d'artifice.

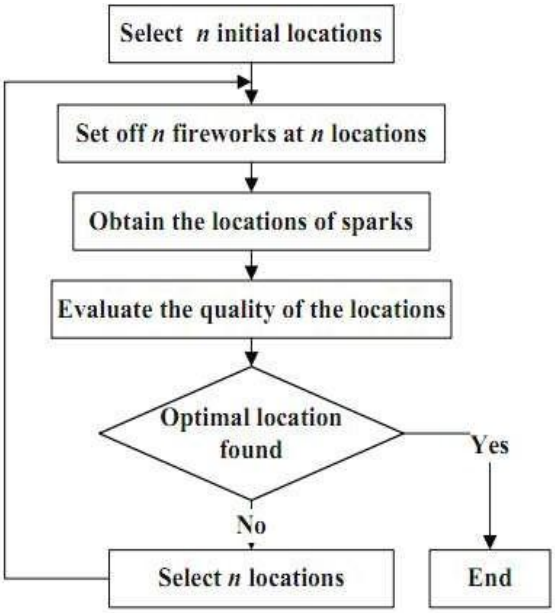


Figure III.1 : Framework standard de l'algorithme des feux d'artifice.

Par la suite, les feux d'artifice explosent et génèrent différents types d'étincelles (Sparks) dans leur espace local.

Enfin, N feux d'artifice candidat sont sélectionnés parmi l'ensemble des candidats (après l'évaluation de leurs qualités), ce qui inclut les étincelles nouvellement générées ainsi que

les N feux d'artifice originaux.

Afin d'assurer la diversité et d'équilibrer la recherche globale et locale, l'amplitude de l'explosion et la population des étincelles d'explosion nouvellement générées diffèrent parmi les feux d'artifice.

6. Définition d'un feu d'artifice

6.1 Un bon feu d'artifice (Good Firework)

Un feu d'artifice avec une meilleure forme physique peut générer une plus grande population d'étincelles d'explosion dans une gamme plus petite, c'est-à-dire avec une petite amplitude d'explosion (voir Figure 6-A).

6.2 Un mauvais feu d'artifice (Bad Firework)

Au contraire, les feux d'artifice ayant une forme physique inférieure ne peuvent générer qu'une population plus petite dans une plage plus large, c'est-à-dire avec une amplitude d'explosion plus élevée (voir Figure 6-B).

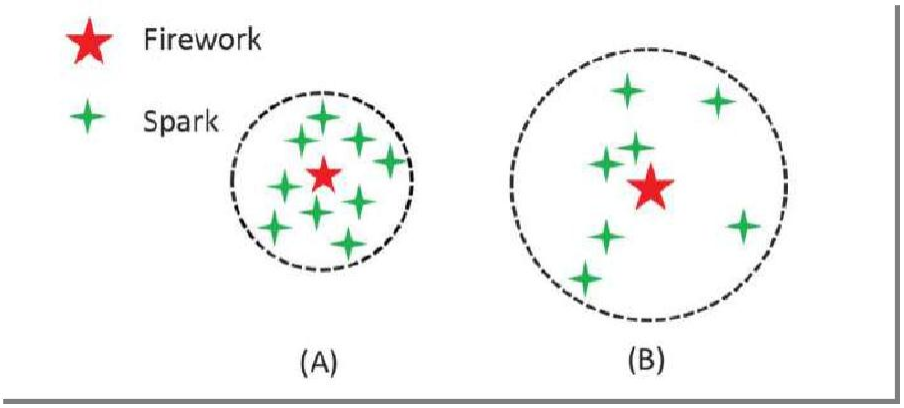


Figure III.2 : Les bonnes et les mauvaises explosions d'un feu d'artifice

A. une bonne explosion. B une mauvaise explosion

.

Cette technique permet d'équilibrer les capacités d'exploration et d'exploitation de l'algorithme. Alors que l'exploration fait référence à la capacité de l'algorithme à explorer différentes régions de l'espace de recherche afin de localiser de bonnes solutions prometteuses, l'exploitation fait référence à la possibilité de mener une recherche approfondie dans une zone plus petite reconnue comme prometteuse [3].

7. Les principaux processus dans l'algorithme des feux d'artifice

7.1 Le processus d'exploration

L'exploration est réalisée par les bons feux d'artifice qui ont une grande amplitude d'explosion, puisqu'ils ont la capacité d'échapper aux minima locaux (voir Figure III.2-A)

7.2 Le processus d'exploitation

L'exploitation est réalisée par les mauvais feux d'artifice qui ont une faible amplitude d'explosion, puisqu'ils renforcent la capacité de recherche locale dans des zones prometteuses (voir Figure III.2 -B).

7.3 Le processus de mutation

Après l'explosion, un autre type d'étincelles est généré sur la base d'une mutation gaussienne des feux d'artifice choisis au hasard. L'idée derrière ceci est d'assurer davantage la diversité de l'essaim.

Afin d'améliorer la lisibilité, les fondateurs de l'algorithme assignent de nouvelles notations aux deux types d'étincelles distinctes: les étincelles d'explosion sont générées par le processus d'explosion, et les étincelles gaussiennes sont générées par une mutation gaussienne [3].

8. Les paramètres principaux de l'algorithme des feux d'artifice

Comme toute méthode d'optimisation, l'efficacité de l'algorithme des feux d'artifice dépend de certains paramètres qui sont:

8.1 L'opérateur d'explosion

Lors de l'initialisation, l'algorithme génère N feux d'artifice de manière aléatoire. Ensuite, les N feux d'artifice génèrent des étincelles par les opérations d'explosion.

L'opérateur d'explosion est un facteur clé dans l'algorithme et joue un rôle très important. En effet, ce dernier inclut la force d'explosion, l'amplitude de l'explosion et l'opération de déplacement .

8.2 La force de l'explosion (le nombre des étincelles) [3]:

Supposons que l'algorithme est désigné pour le problème d'optimisation suivant:

$$\min f(x) \in R, x_{min} \leq x \leq x_{max}$$

Tel que $\mathbf{x= x_1, x_2, \ldots, x_n}$ indique un emplacement dans l'espace potentiel de recherche,

f(x) est une fonction objective, et $\mathbf{x_{min}}$ et $\mathbf{x_{max}}$ désignent les limites de l'espace potentiel. Le nombre des étincelles générés par chaque feu d'artifice x_i est défini par l'équation

suivante:

$$s_i = m \cdot \frac{y_{\max} - f(\boldsymbol{x_i}) + \xi}{\sum_{i=1}^{n} (y_{\max} - f(\boldsymbol{x_i})) + \xi}$$

où **m** est un paramètre qui contrôle le nombre total d'étincelles générées par les N feux d'artifice, $\mathbf{y_{max}}$ = max (f (x_i)) (i = 1, 2, ..., n) est le maximum (le pire) valeur de la fonction objectif parmi les N feux d'artifice, et ξ, qui dénote la plus petite constante dans l'ordinateur, est utilisée pour éviter l'erreur de division par zéro. Pour éviter les effets accablants des feux d'artifice splendides, des limites sont définies pour si, comme montré dans l'équation suivante:

$$\hat{si} = \begin{cases} round\ (a.m)\ si\ Si < am \\ round\ (b.m) si\ Si > bm, \quad a < b < 1 \\ round\ (Si) Sinon \end{cases}$$

Avec a et b sont des paramètres constants.

8.3 L'amplitude de l'explosion [3]:

Contrairement à la conception du nombre d'étincelles, l'amplitude d'un bon feu d'artifice est plus faible que celle d'un mauvais feu d'artifice. L'amplitude d'explosion pour chaque feu d'artifice est définie par l'équation suivante:

Où

$$A_i = \hat{A} \cdot \frac{f(\boldsymbol{x_i}) - y_{\min} + \xi}{\sum_{i=1}^{n} (f(\boldsymbol{x_i}) - y_{min}) + \xi}$$

Â désigne l'amplitude maximale de l'explosion, et

y_{min}=min (f (x_i)) (i=1, 2, . . ., n) est la valeur minimale (la meilleure) de la fonction objectif parmi les N feux d'artifice.

8.4 Génération des étincelles [3]:

Dans le processus d'explosion, des étincelles peuvent subir les effets de l'explosion à partir de z directions aléatoires (z dimensions). Dans l'algorithme des feux d'artifice, nous obtenons le numéro de directions affectées au hasard par la formule suivante:

z = round (d * rand (0, 1)),

Où **d** est la dimensionnalité de l'emplacement x, et rand (0, 1) est une distribution uniforme

sur [0,1].

L'emplacement d'une étincelle du feu d'artifice x_i est obtenu à l'aide de l'algorithme 1.

Algorithm 1. Obtain the location of a spark

Initialize the location of the spark: $\tilde{x_j} = x_i$;
z=$round(d \cdot rand(0,1))$;
Randomly select z dimensions of $\tilde{x_j}$;
Calculate the displacement: $h = A_i \cdot rand(-1,1)$;
for each dimension $\tilde{x}_k^j \in$ {pre-selected z dimensions of $\tilde{x_j}$} **do**
  $\tilde{x}_k^j = \tilde{x}_k^j + h$;
  if $\tilde{x}_k^j < x_k^{\min}$ or $\tilde{x}_k^j > x_k^{\max}$ **then**
    map $\tilde{x}_k^j$ to the potential space: $\tilde{x}_k^j = x_k^{\min} + |\tilde{x}_k^j| \%(x_k^{\max} - x_k^{\min})$;
  end if
end for

8.5 Explosion gaussienne (mutation gaussienne) [3]:

Pour garder la diversité des étincelles, les auteurs de l'algorithme utilisent une autre façon de générer des étincelles selon l'algorithme 2.

La fonction Gaussienne (1, 1) dénote une distribution gaussienne avec la moyenne 1 et l'écart-type 1, est utilisé pour définir le coefficient de l'explosion.

D'après les expériences des auteurs, mˆ étincelles de ce type sont générés dans chaque génération d'explosion.

Algorithm 2. Obtain the location of a specific spark

Initialize the location of the spark: $\hat{x}_j = x_i$;
z=$round(d \cdot rand(0,1))$;
Randomly select z dimensions of $\hat{x}_j$;
Calculate the coefficient of Gaussian explosion: $g = Gaussian(1,1)$;
for each dimension $\hat{x}_k^j \in$ {pre-selected z dimensions of $\hat{x}_j$} **do**
  $\hat{x}_k^j = \hat{x}_k^j \cdot g$;
  if $\hat{x}_k^j < x_k^{\min}$ or $\hat{x}_k^j > x_k^{\max}$ **then**
    map $\hat{x}_k^j$ to the potential space: $\hat{x}_k^j = x_k^{\min} + |\hat{x}_k^j| \%(x_k^{\max} - x_k^{\min})$;
  end if
end for

8.6 L'opération de déplacement [3]:

Au début de chaque génération d'explosion, N emplacements doivent être sélectionnés pour l'explosion des feux d'artifice. Dans l'algorithme de feux d'artifice, le meilleur

emplacement actuel *x**, sur lequel la fonction objectif *f (x*)* est optimale parmi les emplacements actuels, est toujours conservée pour la prochaine génération d'explosion. Après cela, N - 1 emplacements sont sélectionnés en fonction de leur distance à d'autres endroits afin de garder la diversité des étincelles.

La distance entre un emplacement x_i et d'autres emplacements est définie comme suit.

$$R(\boldsymbol{x_i}) = \sum_{j \in K} d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sum_{j \in K} \| \boldsymbol{x_i} - \boldsymbol{x_j} \|$$

Où K est l'ensemble des emplacements actuels des feux d'artifice et des étincelles. Ensuite, la probabilité de sélection d'un emplacement x_i est définie comme suit.

$$p(\boldsymbol{x_i}) = \frac{R(\boldsymbol{x_i})}{\sum_{j \in K} R(\boldsymbol{x_j})} \quad .$$

Pour le calcul de la distance, plusieurs mesures de distance peuvent être utilisées, comme la distance de Manhattan et la distance euclidienne.

9. Les variantes de l'algorithme des feux d'artifice

Depuis l'apparition de l'algorithme des feux d'artifice, il a attiré l'attention des chercheurs pour développer d'autres versions améliorés de l'algorithme.

- Zheng Shaoqiu et al [19] ont proposé un nouvel algorithme amélioré des feux d'artifice, dans leur article, ils ont analysé de manière exhaustive les opérateurs de la version standard et ont souligné les limites de cette version.

- Des implémentations parallèles ont été proposés par différents chercheurs comme dans les travaux de Simone A. Ludwig et al [19].

Conclusion

Il ressort de ce chapitre que les algorithmes des feux d'artifice constituent une nouvelle approche d’optimisation robuste et flexible qui peut être appliquée à une grande variété des problèmes.

Chapitre 4 :
L’ implémentation

1. Introduction

Dans ce chapitre, nous discuterons de la façon de modéliser l'approche proposée en termes de modélisation de solutions, ainsi que de la modélisation de la procédure d'explosion.

Nous discuterons de la manière de mettre en œuvre la solution en abordant le langage de programmation utilisé, la base de données utilisée, ainsi que les mesures d'efficacité utilisées.

2. La modélisation de l'approche

Afin de créer la version discontinue de l'algorithme de feux d'artifice liée au problème d'extension de requête, nous avons besoin d'une modélisation de requête ainsi que de la modélisation d'explosion de feux d'artifice, et pour cela nous devons définir le concept de distance comme deux extensions de la requête et le concept ultérieur de contiguïté Chaque requête étendue, qui est une solution possible à partir de l'ensemble total de solutions, se compose de deux parties, la partie statique, qui se compose des mots de la requête d'origine, et la partie variable ou la partie d'expansion, qui se compose des mots que nous apportons à partir de l'ensemble de mots candidats pour faire partie de la requête étendue et sont obtenus à partir de La source utilisée pour développerla requête, qui dans notre cas est le meilleur des mots extraits des meilleurs documents extraits à l'aide de la requête d'origine. Les solutions initiales sont configurées de manière aléatoire, puis l'efficacité des solutions est évaluée sur la base d'un processus de recherche, et la valeur du premier document récupéré est considérée comme la valeur de la requête étendue.

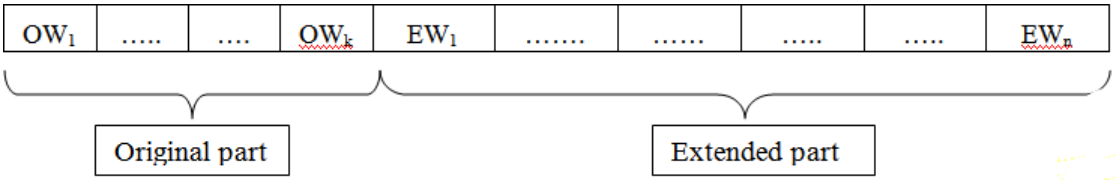


Figure VI.1 : La modélisation de l'approche

Pour la modélisation d'explosion, le nombre de fusées éclairantes n'est pas un problème, mais pourla gamme, il est représenté par le nombre de mots qui ne sont pas en commun, représentant deux solutions, c'est-à-dire que plus les deux solutions sont proches, plus les mots sont communs, et plusles mots sont différents, il est à noter que les mots à changer dans la procédure d'explosion sont choisis au hasard.

3. Le langage Python

Python (prononcé /pi.tɔ̃/) est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse- miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Python est considéré comme l'un des langages de programmation les plus performants en terme de traitement et d'analyse de données textuelles et autres, en raison de ses structures de données qui facilitent le travail sur des textes tels que les dictionnaires, ainsi que de la présence d'un grand groupe de bibliothèques prenant en charge l'analyse de données textuelles et autres tels que NLTk pour notre travail on a utilise la version 3.6 de Python

4. Bibliothèques utilisées

Pour le prétraitement on a utilisé la bibliothèque NLTK. NLTK est une plate-forme leader pour construire des programmes Python pour travailler avec des données de langage humain. Il fournit des interfaces faciles à utiliser pour plus de 50 corpus et ressources lexicales telles que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, la tokénisation, le stemming, le tagging, l'analyse et le raisonnement sémantique, des enveloppes pour les bibliothèques NLP de force industrielle, et un forum de discussion actif.

5. Les packages utilises

- le package collections
- le package pickle
- le package itertools
- le package random
- BeautifulSoup from BS4
- stem from stemming.porter2
- Counter from Collections
- os
- math
- operator
- -- csv

6. Le processus d'évaluation

Un protocole regroupe la description des conditions et des étapes de déroulement d'une

expérience, dans notre cas l'évaluation des systèmes de recherche d'information. La description doit être suffisamment claire afin que l'expérience puis se être reproduite et il doit faire l'objet d'une analyse critique pour détecter les limites afin d'évoluer les systèmes.

Dès les années 1970, un protocole complet d'évaluation de systèmes d'informations documentaire a été conçu et mis en place dans le cadre du projet Smart de Gérald Salton .

Pour évaluer un système de recherche d'informations, il suffira de lui soumettre les questions tests, et de comparer les réponses qu'il fournira aux réponses attendues.

En mesurant la différence entre la réponse du système et la réponse attendue pour obtenir une mesure de qualité sur les performances du système documentaire. Un protocole d'évaluation reposegénéralement sur des collections de tests et des jugements de pertinence.

7. Collection de tests

Pour évaluer le degré de pertinence des réponses d'un système de recherche d'information à une requête, il est nécessaire de connaître l'ensemble des documents pertinents pour une requête donnée. C'est à cette fin que des collections de tests ont été élaborées. Une collection de tests comprend :

Des corpus de documents : une liste de requêtes prédéfinies et des jugements de pertinence.

Requête : c'est une représentation d'un besoin d'information. Il est souhaitable que le protocoled'évaluation prend en compte une requête directement dans la forme où elle a été soumise ausystème, et non un besoin ou une question exprimée sous forme libre et détaillée (éventuellementbeaucoup plus riche d'informations). Ceci restreint d'emblée les objectifs même de l'évaluation,qui ne permettra pas d'effectuer des mesures de performance relatives (entre SRI), si ces SRIn'offrent pas le même langage de requêtes et n'exploitent pas les mêmes enrichissements associésaux documents.

Jugements de pertinence: ils sont manuellement établis et constituent la liste des documents pertinents pour chaque requête. Ils peuvent être portés par l'utilisateur lui-même, ou par un observateur expert 'extérieur' qui se concentrera plus facilement sur la « pertinence ». Ils peuvent prendre des formes booléennes variées (le document est pertinent ou il ne l'est pas) ou plus nuancées, généralement par l'usage de pondérations. L'utilisation de poids ou de pondérations implique la définition d'une échelle de mesure.

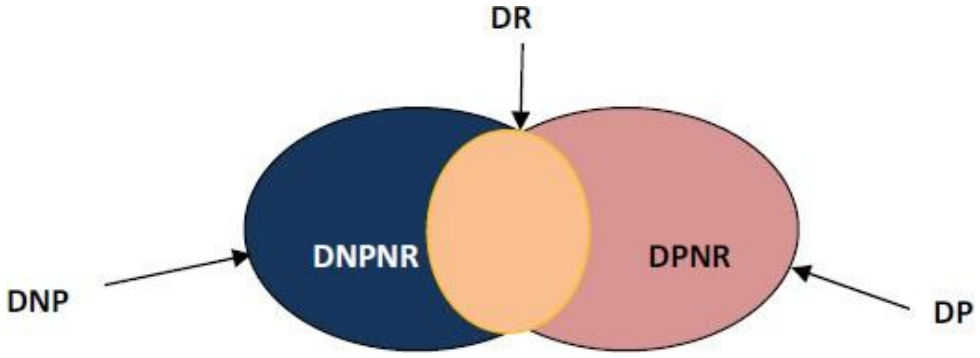
pour notre travail la collection de test est la collection CACM avec 3204 documents et 64 requets

8. Mesures d'évaluation

Tout l'enjeu du processus de recherche d'information est de minimiser la distance entre la pertinence système et la pertinence utilisateur. Plusieurs mesures standards en RI ont été proposées pour évaluer les performances des SRI. Nous nous basons sur les travaux de pour présenter ces mesures.

9. Rappel et Précision

Le rappel et la précision sont deux mesures de base pour évaluer les performances des systèmes de recherche d'information. Le principe de ces deux mesures est basé sur la connaissance à-priori des documents pertinents de la collection d'une part, et d'autre part la partition de l'ensemble des documents restitués par le SRI en deux catégories : documents pertinents et documents non pertinents. La figure 3.1, illustre la partition de la collection de tests pour une requête.



DP : Documents pertinents
DNP : Documents non pertinents
DR : Documents retrouvés
DPNR : Documents pertinents non retrouvés
DNPNR : Documents non pertinents non retrouvés

Figure VI.2 : Rappel et Précision de système

Le Rappel

Cette mesure peut être vue comme une mesure de couverture du système. Elle calcule la capacité du SRI à retrouver les documents pertinents de la collection. Le rappel indique le pourcentage de documents pertinents qui ont été retrouvés par le SRI par rapport à l'ensemble des documents pertinents de la collection.

$$\textbf{Rappel} = \frac{Nombre\ des\ documents\ pertinents\ retrouvés}{Nombre\ des\ documents\ pertinents} = \frac{|DPR|}{|DP|}$$

La Précision

Cette mesure calcule la capacité du SRI à retrouver uniquement les documents pertinents. La précision permet de mesurer la fraction des documents pertinents parmi ceux qui ont été retrouvés par le système.

Un SRI idéal est un système qui restitue tous les documents pertinents (rappel = 1), et tous les documents qu'il retrouve sont pertinents (précision =1) pour la requête de l'utilisateur. En pratique, cet idéal n'est jamais atteint puisque ces deux quantités évoluent en sens inverse. Intuitivement, si on augmente le rappel en retrouvant plus de documents pertinents, et si on diminue la précision en retrouvant aussi plus de documents non pertinents. Inversement, une plus grande précision risque de rejeter des documents pertinents diminuant ainsi le rappel [08].

$$\textbf{Précision} = \frac{Nombre\ des\ documents\ pertients\ retrouvés}{Nombre\ des\ documents\ retrouvés} = \frac{|DPR|}{|DPR \cup DNPR|}$$

Tests et résultats

Pour des raisons du temps on a testé que la requête N 1 et voici les résultats de test

1,CACM-1605,0.014705882352941176,0.02040816326530612,R
1,CACM-1277,0.014492753623188406,0.02040816326530612,NR
1,CACM-1347,0.014285714285714285,0.02040816326530612,NR
1,CACM-2188,0.014084507042253521,0.02040816326530612,NR
1,CACM-1299,0.013888888888888888,0.02040816326530612,NR
1,CACM-1298,0.0136986301369863,0.02040816326530612,NR
1,CACM-0366,0.013513513513513514,0.02040816326530612,NR
1,CACM-0200,0.013333333333333334,0.02040816326530612,NR
1,CACM-1192,0.013157894736842105,0.02040816326530612,NR
1,CACM-0717,0.012987012987012988,0.02040816326530612,NR
1,CACM-1179,0.01282051282051282,0.02040816326530612,NR
1,CACM-2868,0.012658227848101266,0.02040816326530612,NR
1,CACM-1571,0.0125,0.02040816326530612,NR
1,CACM-1190,0.012345679012345678,0.02040816326530612,NR
1,CACM-0347,0.012195121951219513,0.02040816326530612,NR
1,CACM-1642,0.012048192771084338,0.02040816326530612,NR
1,CACM-0228,0.011904761904761904,0.02040816326530612,NR
1,CACM-1189,0.011764705882352941,0.02040816326530612,NR
1,CACM-1283,0.011627906976744186,0.02040816326530612,NR
1,CACM-2724,0.011494252873563218,0.02040816326530612,NR
1,CACM-2189,0.011363636363636364,0.02040816326530612,NR
1,CACM-1187,0.011235955056179775,0.02040816326530612,NR
1,CACM-1191,0.011111111111111112,0.02040816326530612,NR

1,CACM-1275,0.01098901098901099,0.02040816326530612,NR
1,CACM-1228,0.010869565217391304,0.02040816326530612,NR
1,CACM-1157,0.010752688172043012,0.02040816326530612,NR
1,CACM-1303,0.010638297872340425,0.02040816326530612,NR
1,CACM-1360,0.010526315789473684,0.02040816326530612,NR
1,CACM-1752,0.010416666666666666,0.02040816326530612,NR
1,CACM-0624,0.010309278350515464,0.02040816326530612,NR
1,CACM-1302,0.01020408163265306,0.02040816326530612,NR
1,CACM-1284,0.010101010101010102,0.02040816326530612,NR
1,CACM-0277,0.01,0.02040816326530612,NR

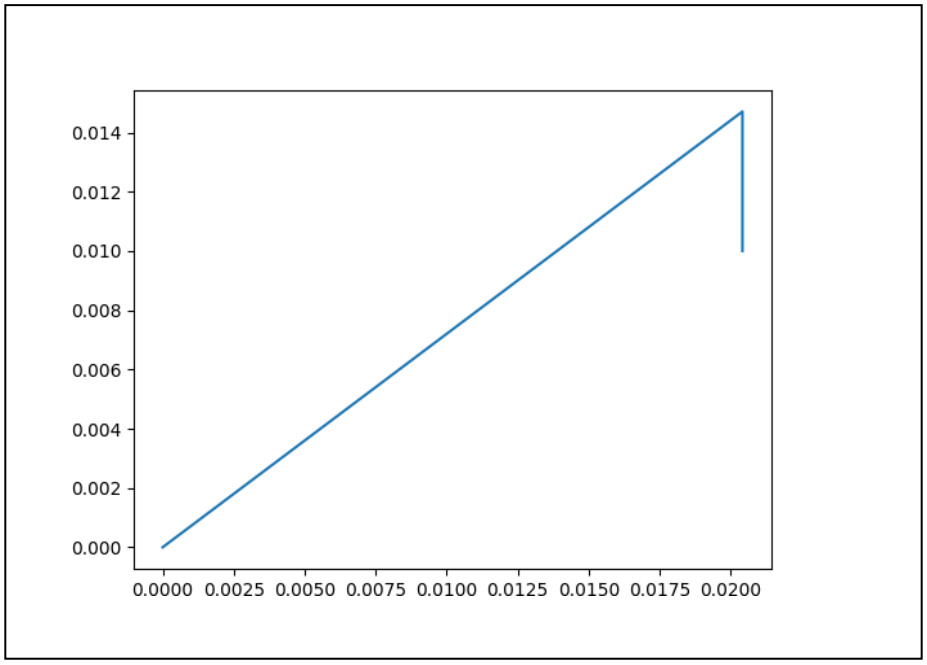


Figure VI.3 : La courbe de pour la première requête

Conclusion

Dans ce chapitre, nous avons vu comment modéliser le modèle proposé et comment le mettre en œuvre. Nous avons également vu les méthodes et l'environnement nécessaires pour évaluer le système proposé, ainsi que les mesures d'efficacité utilisées, mais par manque de temps, nous avons pu inclure uniquement les résultats obtenus pour la première requête uniquement. Nous espérons pouvoir terminer les travaux bientôt.

Conclusion générale

L'algorithme de feux d'artifice est l'un des algorithmes d'intelligence en essaim qui a prouvé son efficacité dans la résolution de problèmes de recherche de solutions optimales, nous l'avons donc adopté comme approche pour résoudre une nouvelle représentation du problème d'extension de requête. Étant donné que les valeurs des variables dans le problème posé ne sont pas des nombres réels, cela pose un défi pour adapter l'algorithme au problème. Cependant, les résultats obtenus justifient l'effort à fournir, il ne reste plus qu'à travailler sur l'ajustement des paramètres, ainsi qu'à travailler pour réduire le temps d'exécution

Il convient de noter que la longueur du temps d'exécution est due au fait que le processus d'évaluation de la fonction objectif est une procédure de recherche et d'arrangement dans le code et non à partir du noyau des propriétés de l'algorithme Pour résoudre ce problème, on peut recourir au parallélisme

Bibliographie

[1] Mostepha, R : Résolution de problèmes d'optimisation combinatoire par systèmes artificiles auto- organisés. Thèse de magister, Université Mentouri de Constantine ,2008.

[2] Lynda Tamine : Optimisation de requêtes dans un système de recherche d'information approche basee sur l'exploitation de techniques avancees de l'algorithmique genetique. Thèse de Doctorat, Université PAUL SABATIER DE TOULOUSE21/12/2000

[3] Palpant M. : Recherche exacte et approchée en optimisation combinatoire : schémas d'intégration et applications. Thèse de Doctorat, Université d"Avignon, 2005.

[4] Aissa Boulmerka, « Adaptation des métaheuristiques à l'ordonnancement hors-ligne d tâches temps réel à contraintes strictes en environnement monoprocesseur », Mémoire de magister en informatique, 20 Avril 2009.

[5] Omessaad, H : Contribution au développement de méthodes d'optimisation Stochastiques application à la conception des dispositifs électrotechniques, Thèse de Doctorat, Université De Lille France 2003.

[6] 3ieme Conference francophone de modelisation et simulation "conception, analyse et gestion des systemes industriels" MOSIM'01 – du 25 au 27 avril 2001 - Troyes (France)

[7] Hiteshwar K A, Akshay De. «Query expansion techniques for Information Retrieval: A survey ».Article (Information Processing & Management). 01 Août 2015.

[8] Alain Hertz, 'L'optimisation Combinatoire', École Polytechnique, Canada, 2006

[9] Rachid Chelouah, 'L'optimisation combinatoire', INA Institut d'informatique appliquée, Suisse, 2003.

[10] http://khayyam.developpez.com/articles/algo/genetic/

[11] Souquet Amédée « Algorithme génétique » Radet Francois-Gérard

[12] https://fr.wikipedia.org/wiki/M%C3%A9taheuristique#Applications

[13] B Audeh. « Reformulation Sémantique des requêtes par la recherche d'information adhoc sur le web ». Thèse de doctorat à l'Ecole nationale supérieur des Mines de Saint-Etienne.6 Mars 2015

[14] Hiteshwar K A, Akshay De. «Query expansion techniques for Information Retrieval: Asurvey».Article (Information Processing & Management). 01 Août 2015 Hiteshwar K A, Akshay De. «Query expansion techniques for Information Retrieval: Asurvey ».Article (Information Processing & Management). 01 Août 2015

[15] bissan audeh : reformulation semantique des requettes pour la recherche d'information ad hoc sur le web : Thèse de Doctorat, l'École Nationale Supérieure des Mines de Saint-Étienne , 09/09/2014.