UNIVERSITE DE DROIT, D'ECONOMIE ET DES SCIENCES D'AIX-MARSEILLE III

FACULTE DES SCIENCES ET TECHNIQUES DE SAINT-JERÔME

CENTRE DE **R**ECHERCHE **R**ETROSPECTIVE DE **M**ARSEILLE

N° attribué par la bibliothèque

/0/2/A/I /X/3/0/0/3/2/

OPTIMISATION DE LA REPRESENTATION DE GRAPHES PAR APPROCHE HYBRIDE DETERMINISTE ET STOCHASTIQUE

T H E S E

pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITE DE DROIT, D'ECONOMIE ET DES SCIENCES D'AIX-MARSEILLE

Discipline : **SCIENCES DE L'INFORMATION ET DE LA COMMUNICATION**

présentée et soutenue publiquement par

Jean-Pierre GHILARDI

Le 10 juillet 2002

Directeur de thèse : Luc Quoniam

J U R Y

- M Tony Tschaegle — Professeur, Université de Nice – Sophia Antipolis
- M. Thierry Lafouge — Maître de Conférence, Université Lyon I
- M. Henri Dou — Professeur, Université de Marseille III
- M. Luc Quoniam — Professeur, Université de Toulon
- M. Eric Boutin — Maître de Conférence, Université de Toulon

A N N E E : 2 0 0 2

SOMMAIRE

Table des illustrations

INTRODUCTION

Le Centre de Recherche Rétrospective de Marseille s'est spécialisé depuis de nombreuses années en veille stratégique.

Le haut niveau d'expertise atteint par le centre s'est notamment traduit par la mise au point d'une chaîne complète d'outils de traitement de l'information bibliographique.

Ces outils sont destinés non seulement à faire progresser la recherche dans le domaine des sciences de l'information, mais aussi à promouvoir des méthodes innovantes de traitement de l'information auprès des principaux acteurs du domaine.

Les techniques mises en œuvre à l'occasion de ces travaux ont notamment fait appel à l'exploitation de représentations graphiques de relations reliant des entités : les graphes.

Les graphes constituent un moyen visuel simple d'appréhender l'organisation, les relations entre les entités, sous-jacentes au phénomène à étudier.

L'objectif de ce travail est d'étudier et de définir les moyens à mettre en œuvre pour aboutir de manière automatique à une expression graphique organisée des graphes et ainsi d'en faciliter l'exploitation par le plus grand nombre et ce sans nécessiter une grande expérience des techniques de traitement mises en œuvre.

L'orientation qui a donc été retenue dans le cadre de ce travail est une approche générique de la problématique, aboutissant à la définition d'un processus de traitement susceptible de prendre en compte une grande diversité de critères esthétiques, plutôt que la recherche du critère esthétique adapté à une problématique donnée ou à un utilisateur donné.

C'est pourquoi, la définition d'un graphe qui a été retenue est la plus générale qui puisse être donnée, elle se résume à sa plus simple expression : « des sommets reliés par des arêtes ». Il en va de même pour le critère d'esthétisme qui est le nombre de croisements d'arêtes que l'on va tenter de minimiser.

La complexification du problème (graphes valués, distance inter sommets, …) pourra être soit déportée vers la définition du critère d'esthétisme finalement

retenu, soit comme cela est évoqué en conclusion, prise en charge par un autre processus prenant en compte des critères plus subjectifs.

1 PROBLEMATIQUE

Nous observons, depuis quelques années, une forte évolution des technologies liées à l'information.

Les télécommunications :

Les supports de transmission de l'information sont en constante progression. Les débits de transfert et donc les volumes d'informations échangés ne cessent de croître. L'offre ADSL qui permet à chaque particulier d'accéder par liaison rapide au réseau informatique Internet est déjà disponible pour tous dans les plus grandes agglomérations françaises. Il en va de même pour les connexions haut-dédits des réseaux locaux d'entreprise via des protocoles ATM ou Frame Relay. Les supports radio ne sont pas en reste avec la mise en œuvre de boucles locales radio qui supportent déjà des vitesses de transmission de l'ordre de deux millions de bits par seconde ou encore la nouvelle norme de téléphonie mobile UMTS donnant accès depuis n'importe quel point du territoire à des taux de transfert équivalents.

Les supports de stockage de masse :

Parallèlement, les supports de stockage de l'information ont vu aussi leur capacité croître de manière exponentielle. Les disques durs exploités sur des unités légères de traitement ont des capacités exprimées en dizaines de milliard de caractères. Il est inconcevable d'acquérir aujourd'hui un ordinateur personnel sans support d'archivage de type cédérom, D.V.D, D.A.T., D.L.T ou autres L.T.O. qui autorisent aussi la diffusion à grande échelle de bases de données stabilisées susceptibles d'être exploitées à l'aide d'outils adaptés.

L'Internet, un autre bouleversement :

L'Internet qui est le réseau des réseaux, banalise par le biais d'outils d'exploration ergonomiques ainsi que par la mise en œuvre de protocoles informatiques et de télécommunication standards, l'accès à la grande majorité des systèmes informatiques de la planète. Il offre un accès instinctif à l'information favorisant ainsi sa diffusion au plus grand nombre.

Les internautes bénéficient ainsi d'avancées technologiques qui restituent un certain confort d'utilisation et offrent au particulier de nouvelles fonctionnalités accessibles depuis un micro-ordinateur. Parmi celles-ci on peut citer l'accès et l'exploitation de bases de données en ligne, la téléphonie, la musique, la vidéo, la communication écrite en ligne ou non, la visioconférence à deux ou à plusieurs, …

La technologie permet de prendre en compte plus de besoins de l'utilisateur :

Avant l'ère de l'Internet, les banques de données implantées sur des systèmes informatiques centraux autorisaient le stockage de l'information massive dans un but essentiellement d'archivage et de recherche de l'information. L'accès à cette information structurée et codifiée étaient réservé à des spécialistes, formés à des langages et à des outils d'interrogation propriétaires complexes à mettre en œuvre.

Aujourd'hui l'accès à l'information est facilité et instinctif. Le rapatriement de celle-ci sur un ordinateur local, en vue de la retravailler est intégré aux explorateurs Internet (automatisation des échanges de fichiers par le biais des protocoles ftp ou http) avec des formats de données directement exportables vers des tableurs ou vers des gestionnaires personnels de bases de données intégrant des langages d'interrogation structurés enrichis d'assistants de requêtage.

La constitution de bases de données personnelles ne pose plus de difficulté, l'utilisateur final (le décideur) est donc à même de créer ses bases de données personnelles sans la nécessité d'acquérir une grande expertise informatique.

La difficulté maintenant, quel que soit notre domaine d'intérêt, est la masse d'information à laquelle nous sommes confrontés. Comment aborder cette masse d'information, comment l'exploiter, comment avoir une vision simplifiée mais non restrictive de la nature des informations dont on dispose, quels traitements appliquer pour synthétiser et extraire l'information pertinente ?

Cela implique la mise en œuvre d'outils de traitement de l'information autorisant une certaine automatisation. Ce domaine est un des axes de recherche du Centre de Recherche Rétrospective de Marseille. L'objectif est de pré-traiter cette information dans le but de mieux en apprécier le contenu.

Comme nous l'explique E. Boutin [Boutin-99], « Une recherche d'information spécifique renvoie souvent à une « information massive » que l'entendement humain a du mal à appréhender pleinement », mais aussi que « Trop d'information tue l'information », d'où la nécessité de développements théoriques importants dans le domaine, initialement appliqués aux informations structurées obtenues depuis les bases de données bibliographiques ou de brevet, maintenant intégrant aussi la dimension Internet.

Ces traitements théoriques débouchent sur la définition et la réalisation d'outils de traitement automatique de l'information. Véritables assistants de l'utilisateur,

ils permettent de mieux rechercher l'information, la sélectionner puis enfin l'analyser pour en tirer bénéfice.

Les domaines d'application sont très variés, on peut citer comme exemple l'analyse des bases de données de brevets évoquées précédemment. Elle permet aux entreprises d'établir une stratégie de développement en apportant des réponses aux questions suivantes : comment évolue le marché, que savent faire les concurrents, quelles sont les orientations qu'ils prennent, quelles orientations prendre, quelles sont les diversifications possibles, …

Le traitement de cette information de masse, dont sont submergées les entreprises, a mis en évidence la nécessité de former des spécialistes, capables de traiter l'information massive pour produire une information à forte valeur ajoutée, destinée principalement aux preneurs de décisions : l'objectif étant de structurer et de rationaliser le traitement des données brutes pour produire des données élaborées, pertinentes et à forte valeur ajoutée.

Ces dernières années ont vu apparaître de nouveaux produits orientés « Exploitation de l'information massive » qui en favorisent la compréhension. Une nouvelle spécialité informatique a vu le jour : le « Datamining ». Cette approche s'est traduite par une évolution de la structure des informations maintenues au sein des systèmes de gestion de bases de données.

Initialement, structurer une base de données correspondait à la prise en compte du compromis volume de données stocké, temps de recherche de l'information massive. L'objectif de méthodes telles que « Merise » était notamment de modéliser l'information sous forme d'entités et de relations dans le but de retrouver celle-ci de manière rapide, et sous une forme adaptée à des traitements définis à l'avance durant une phase de spécification informatique. Pour cela l'information massive était stockée conformément à un modèle de données orienté : « manipulation de l'information ».

Les évolutions technologiques ont permis d'intégrer la dimension : compréhension, vision synthétique de l'information au travers d'un deuxième modèle de données maintenu dans de nouveaux systèmes de gestion de bases de données. Ce modèle orienté « analyse de l'information », favorise l'application d'opérateurs de traitement de l'information.

De nombreux produits de « DATAMINING » sont maintenant disponibles sur le marché, ils sont essentiellement orientés traitement de l'information numérique, par application d'opérateurs statistiques basiques. L'information massive est

stockée dans des bases de données où deux modèles conceptuels de données cohabitent.

Ce ne sont plus seulement les tailles de stockage et les temps de traitement qui sont favorisés, mais aussi l'application de nouveaux opérateurs statistiques, s'appuyant sur des pseudo-modèles conceptuels de données, à savoir des univers de données qui constituent de véritables filtres d'accès aux informations stockées dans des entrepôts de données.

Pour l'instant ces opérateurs sont relativement basiques, ils sont essentiellement statistiques, appliqués à des données numériques, ou à des caractéristiques numériques de données textuelles. Ils sont intégrés dans des interfaces conviviales qui permettent aux décideurs, après une brève formation, d'effectuer une première analyse de l'information massive contenue dans les bases de données de leurs entreprises.

Aller plus loin dans l'analyse de l'information de masse dont on dispose suppose la mise en œuvre de démarches automatiques d'aide à la construction de sens, une de ces méthodes repose sur la construction automatique de réseaux ou de graphes, ce qui est le sujet de ce travail.

Un exemple d'application des représentations de l'information sous forme de réseaux est la formalisation graphique des liens hypertextes qui autorisent l'exploration d'un site Internet. L'objectif est de fournir à l'utilisateur une représentation la plus claire possible de la navigation qu'il est susceptible de suivre pour aboutir à l'information recherchée. Des critères d'esthétisme relatifs à la représentation graphique d'un réseau de pages d'un site Web peuvent être définis comme par exemple la minimisation du nombre de croisements des arêtes constituant le réseau des pages.

Un autre exemple d'application de la représentation de l'information sous forme de graphes découle de l'interrogation des banques de données internationales qui fournissent une information élaborée, sous la forme de notices bibliographiques.

Même si les codifications exploitées par les différents producteurs de banques de données sont spécifiques, il n'en reste pas moins que la structure de celles-ci reste relativement constante : champs auteurs, champs pays, champs mot clés, … Après une phase d'harmonisation des sources de données, là encore l'exploitation de représentations sous forme de réseaux pourra être bénéfique.

Ces exemples sont développés plus loin.

Nous serons par exemple à même de mettre en évidence sous forme de graphes exploitables par une lecture directe, les liens de collaboration qui peuvent exister entre des laboratoires de recherche et des entreprises privées, des fournisseurs et des clients, des collaborations d'entreprises, …

Il est important de souligner que la représentation graphique sous forme de réseaux encore appelés graphes n'est pas réductrice en terme de quantité d'information restituée, elle en est une simple représentation graphique, qui, si elle est suffisamment claire, peut être exploitée immédiatement par l'utilisateur. Il n'y a pas nécessité d'appliquer une gymnastique mentale pour interpréter un résultat découlant de transformations complexes appliquées au corpus d'informations initial.

1.1 LES RESEAUX ET LEURS APPLICATIONS

Si les réseaux constituent un moyen de représenter cartographiquement une information structurée, ils se sont aussi avérés une alternative séduisante aux méthodes classiques d'analyse de données : ils facilitent la compréhension de certains phénomènes.

En effet, si la théorie sous-jacente à la construction et à l'organisation des graphes est complexe et élaborée, il n'en reste pas moins que les résultats obtenus sont susceptibles d'être observés par lecture directe.

Le lecteur est soulagé d'un investissement lourd d'interprétation de résultats et peut se consacrer pleinement à l'exploitation de celui-ci et ce sans formation préalable.

Les réseaux permettent de modéliser un grand nombre de phénomènes réels, de mieux les comprendre et ce essentiellement par optimisation de la représentation des interactions entre les différentes entités constituant le phénomène.

Ils ont donné lieu à de nombreux développements :

- Traitements de l'information documentaire, les thésaurus ;
- Bibliométrie, les graphes de paire ;
- Maintenance des sites Web ;
- Méthodologie informatique, les modèles conceptuels de Merise ;
- Analyse de données, les graphes de liaison inter-classe ;
- Sciences sociales, les réseaux sociaux ;
- Etc…

Les réseaux permettent donc de représenter de manière simple et immédiatement compréhensible, les relations pouvant exister entre des entités.

Ils sont constitués au minimum d'entités représentées par des points, des cercles, ou encore des zones de texte, reliées par des segments de droites.

Ils peuvent être enrichis par des informations complémentaires :

- Les entités peuvent être liées plus ou moins fortement, le réseau pourra alors être valué, les arêtes pourront avoir une épaisseur modulable ou une longueur modulable.

- Les relations liant les entités pourront être univoques, le réseau sera alors orienté, le sens de circulation étant matérialisé par des flèches.
- Les entités pourront avoir une importance plus ou moins grande, impactant sur la taille des points les représentant ou le contenu des textes rattachés.

1.2 APPLICATION DES RESEAUX AUX SCIENCES DE L'INFORMATION

1.2.1 Traitement de l'information documentaire, les thésaurus

Le traitement de l'information documentaire dans sa composante indexation et recherche de l'information implique une maîtrise parfaite non seulement du langage d'interrogation des bases de données bibliographiques, mais aussi de l'organisation des termes exploités pour indexer les références qui viennent enrichir continuellement ces bases de données.

Des outils documentaires comme les thésaurus ont ainsi été développés notamment pour rationaliser le vocabulaire utilisé : ils autorisent ainsi une indexation contrôlée des références bibliographiques.

Si les thésaurus ainsi créés facilitent l'utilisation et structurent ce vocabulaire, ils permettent aussi une certaine convergence de la vision que peuvent avoir différents chercheurs sur un sujet de recherche.

Il n'en reste pas moins que cet outil incontournable reste très coûteux à créer mais aussi à maintenir. En effet les thésaurus ne sont pas créés une fois pour toute durant la phase de définition d'une base de données, mais au contraire ils sont très évolutifs de manière à intégrer en permanence les nouveaux termes du vocabulaire scientifique.

Les thésaurus participent à la définition de relations entre les termes relatifs à chaque domaine scientifique et technique. S'ils intègrent des notions relatives aux termes exploités, comme les termes descripteurs, les termes équivalents, …, ils intègrent aussi la notion de relations entre ces termes décrite entre-autres par Jacques Chaumier [Chaumier-90].

1.2.1.1 Thésaurus, termes et relations

Différentes types de relations ont donc été définies, elles permettent d'organiser les liens existant entre les différents types de termes constituant un thésaurus qui sont notamment les suivants :

- *Les descripteurs* sont exploités pour indexer les documents et les retrouver à l'aide de requêtes codifiées par un langage d'interrogation spécifique, ils sont normalisés et se rapportent à un concept spécifique. Ils sont formés d'un mot ou d'une expression.
- *Les termes équivalents* sont constitués des synonymes linguistiques qui correspondent avec exactitude à la même notion que le descripteur auquel ils se rattachent et des synonymes documentaires qui sont exploités pour regrouper, autour des termes descripteurs, les termes considérés comme voisins et ce même s'ils ont une signification sémantique différente.
- *Les mots outils* sont des descripteurs sans signification particulière s'ils sont employés seuls, ils doivent donc être employés en combinaison avec d'autres descripteurs dont ils affinent le sens.
- *Les infra concepts* sont des éléments lexicaux qui accolés à certains descripteurs en modifient le sens.

Ces différents types de termes sont liés par des relations qui participent à créer l'organisation régissant les thésaurus, ces relations sont notamment :

- *Les relations d'équivalence* connectent les synonymes aux descripteurs, elles sont ainsi exploitées pour convertir les termes équivalents en descripteurs et permettent de retrouver pour un descripteur donné, l'ensemble des termes équivalents qui s'y rattachent.
- *Les relations hiérarchiques* permettent d'organiser les descripteurs entre-eux en détaillant quels concepts englobent d'autres concepts. Elle peuvent être représentées graphiquement sous forme arborescente quand elles sont mono-hiérarchiques, c'est à dire quand un terme fils n'est lié qu'à un seul terme père. Leur représentation graphique peut devenir plus ardue dans le cas contraire (polyhiérarchie).
- *Les relations associatives* signalent les analogies qui peuvent exister entre deux termes descripteurs. Elles participent à complexifier

encore les expressions graphiques des thésaurus, excepté quand elles se substituent à une relation polyhiérarchique.

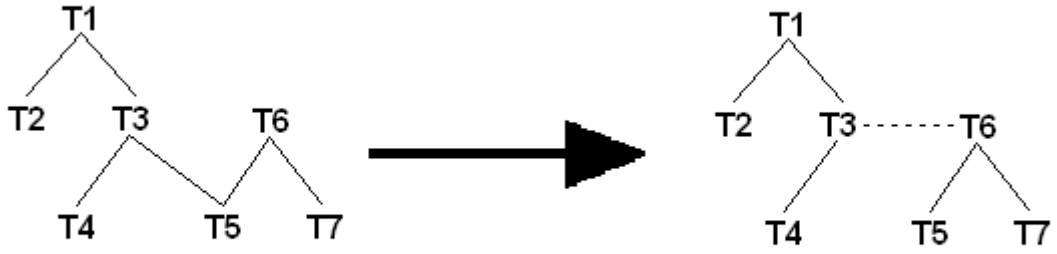


Figure 1-1 : Relations associatives.

- *Les relations de définition*, ces relations qui explicitent un terme descripteur n'ont que peu d'impact sur la complexification des représentations graphiques des thésaurus.
- *Les relations catégorielles* impliquent le regroupement de termes descripteurs en fonction de caractéristiques communes. Elles aussi peuvent engendrer une complexification de l'expression graphique d'un thésaurus.

1.2.1.2 La représentation graphique d'un thésaurus

Les thésaurus sont donc des outils incontournables dans l'interrogation des bases de données, leur complexité induite par le volume des termes qui les composent, la nature et le nombre de relations qu'ils intègrent ainsi que le nombre de domaines scientifiques qu'ils sont susceptibles de couvrir (macrothésaurus) ont poussé les concepteurs de thésaurus à définir des outils graphiques de représentation dont les principales composantes sont les schémas et les tableaux graphiques :

- *Les schémas graphiques* représentent sous forme de flèches les relations sémantiques existant entre les termes descripteurs qui eux sont représentés dans des zones de textes. Des informations complémentaires peuvent êtres ajoutées aux graphes comme la numérotation des descripteurs, les relations de définitions, … Il s'agit en fait de réseaux orientés, qui sous leur forme la plus simple sont de

type arborescents mettant ainsi en évidence les termes descripteurs génériques (les pères). Ils peuvent être complexifiés par extension du périmètre de recherche ou par ajout d'autres types de relations.

- *Les tableaux graphiques* qui sont exploités pour représenter les thésaurus de volume important constituent aussi une expression graphique sous forme de réseau, les zones de texte comprennent alors l'ensemble des termes descripteurs fils rattachés aux termes descripteurs pères du niveau en cours de représentation.

Ces modes de représentation sous forme de réseaux sont nécessaires à la maintenance ainsi qu'à l'exploitation des thésaurus, étant l'expression graphique des thésaurus qu'ils représentent, ils sont par définition évolutifs et doivent pouvoir être reconstruits automatiquement à partir des structures d'enregistrement physique sous-jacentes.

La problématique étant identique à celles citées plus haut : il faut ici aussi pouvoir positionner graphiquement et de manière optimale les différents termes composant la portion de thésaurus concernée par une recherche bibliographique et formant un réseau de relations liant ces termes.

L'apport de ces types de représentation ne constitue pas seulement une amélioration de la perception de la structure d'un thésaurus dans une phase d'enrichissement, ils permettent aussi, lors d'une recherche, de mieux s'orienter dans le méandre des concepts relatifs au domaine scientifique traité et participent donc à améliorer l'efficacité d'une recherche bibliographique.

1.2.2 Bibliométrie

A. Pritchard [Pritchard -69] donne la définition suivante de la bibliométrie « application de méthodes mathématiques et statistiques aux livres et aux médias de communication », cette définition est complétée par la définition fournie par H. Rostaing [Rostaing-93] qui précise que la bibliométrie est « l'application de méthodes statistiques ou mathématiques sur des ensembles de références bibliographiques ».

Certaines de ces méthodes statistiques ou mathématiques ont été développées et commentées dans le cadre du travail de E. Boutin [Boutin-99], qui de plus, décrit l'organisation des références bibliographiques couramment observée sous la forme des quatre définitions qui sont reprises ci-dessous :

1.2.2.1 Corpus

Les données de départ non traitées sont appelées corpus à analyser (ensemble structuré de données massives). Le corpus désignera suivant le cas l'ensemble des notices téléchargées depuis une banque de données, un ensemble de questionnaires saisis lors d'une enquête, l'information récupérée suite à l'utilisation d'un moteur de recherche ou d'un agent intelligent sur Internet.

1.2.2.2 Référence

Pour pouvoir donner lieu à un traitement automatique, tout corpus doit être décomposé en un certain nombre de références (découpage identitaire du corpus). La référence désignera selon le cas une notice bibliographique du corpus, un questionnaire renseigné, un site Internet renvoyé par un moteur de recherche, ...

1.2.2.3 Champs

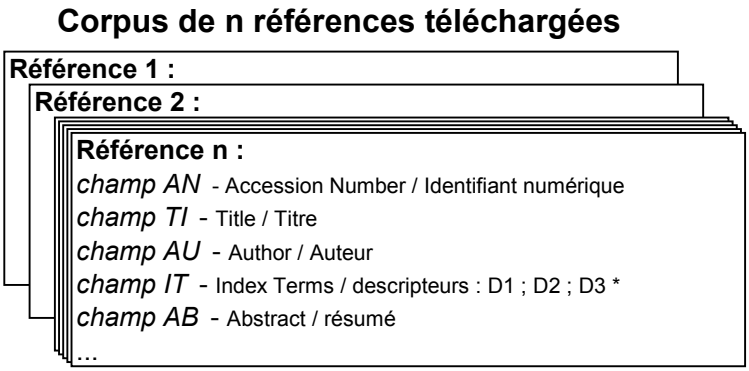
Chaque référence est elle même structurée en un ou plusieurs champs (découpage thématique du corpus). Dans le cas de l'analyse d'un ensemble de brevets, le nom de l'inventeur du brevet de référence est un champ. Dans le cas du traitement d'enquêtes, à chaque question renseignée du questionnaire sera associé un champ. Dans le cas de traitements appliqués à Internet, le résumé du site ou son titre sont deux exemples de champs. Le caractère invariant des champs sur toutes les références permet d'obtenir un corpus homogène et structuré ce qui rend possible des analyses automatiques en ne retenant qu'un ou plusieurs champs.

1.2.2.4 Modalités

Les valeurs possibles d'un champ sont appelées modalités (ensemble des formes que peut prendre un champ) du champ. C'est le niveau le plus fin de l'analyse. La modalité est l'unité d'information élémentaire du corpus. On distingue à ce niveau les champs unimodaux des champs multimodaux. Les premiers sont renseignés par une information unique : le champ « Titre » d'une notice brevet est un champ unimodal dans la mesure où à chaque brevet est associé un titre et un seul. Les champs multimodaux sont composés de plusieurs modalités présentes simultanément. Dans le cas du traitement d'information provenant d'Internet, le champ qui s'intéresse au nom des liens externes d'un

site vers les autres est un champ multimodal, un site pouvant renvoyer à plusieurs autres.

Ces définitions peuvent être synthétisées par la figure suivante :



* ici le champ IT est multimodal.

Figure 1-2 : Structure des références bibliographiques.

1.2.2.5 Structuration de l'information

L'application de méthodes statistiques ou mathématiques sur des ensembles de références bibliographiques implique la mise en œuvre de traitements automatisés de l'information et donc la nécessité de disposer d'une information non seulement sous forme électronique, mais aussi sous forme structurée.

Nous avons évoqué plus haut l'application des représentations réseaux dans le cadre de l'exploitation de thésaurus permettant d'indexer divers documents :

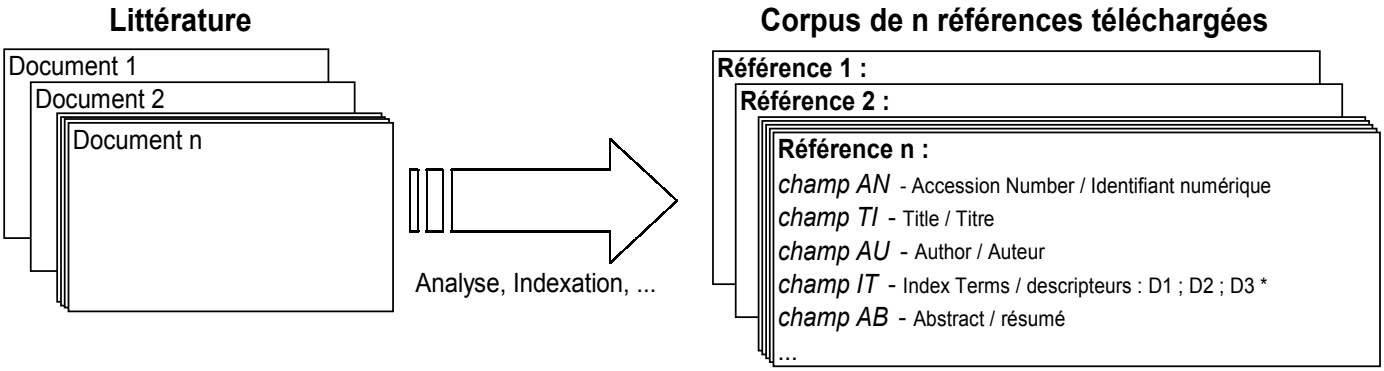


Figure 1-3: Processus de création des bases de données bibliographiques.

Ce processus pris en charge par les producteurs de références bibliographiques qui à partir d'une information primaire génèrent une information secondaire permet ensuite à des serveurs de proposer via des langages d'interrogation spécifiques des corpus de références bibliographiques structurées sur lesquelles vont s'appuyer les analyses bibliométriques.

Les références ainsi téléchargées constituent une information massive qui ne va pas pouvoir être exploitée telle quelle comme support de décision, mais va devoir subir un certain nombre de traitements en vu de produire une information pertinente exploitable par le décideur :

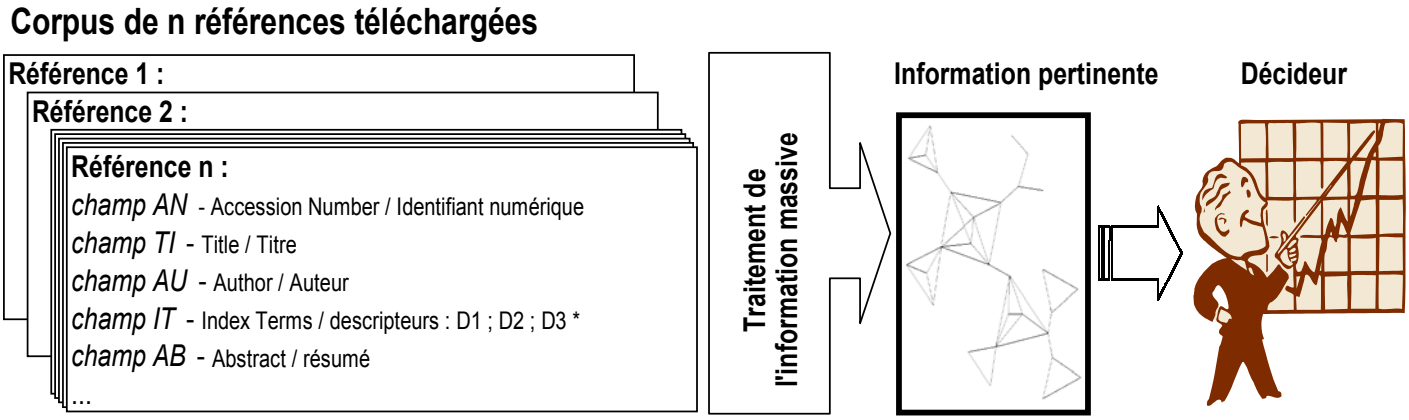


Figure 1-4 : Traitement de l'information massive.

1.2.2.6 Traitement de l'information massive

Comme nous l'avons vu plus haut, des traitements automatisés ne pourront être mis en œuvre que dans la mesure ou l'information massive aura un caractère structuré, en plusieurs champs, eux-mêmes structurés en une ou plusieurs modalités, elles-même faisant appel à des tables de transcodage ou non.

Ces traitements automatisés correspondent à des traitements successifs qui s'enchaînent et dont le premier est constitué le plus souvent par des opérations d'extraction des champs concernés par l'étude ainsi que par des opérations de comptage et de reformatage des modalités de ces différents champs.

Les structures de données ainsi constituées sont alors susceptibles d'être exploitées pour donner lieu à une représentation réseau.

La première matrice construite est une matrice dite de présence-absence :

	Modalité 1	…	Modalité j	…	Modalité n
Référence 1					
…					
Référence i			M(i,j) = 1 si la modalité j est présente		
…				dans la référence i, sinon 0	
Référence m					

Figure 1-5 : Matrice de présence-absence.

Cette matrice va pouvoir être transformée soit en matrice des modalités, soit en matrice des références.

Ces matrices carrées qui vont autoriser une représentation sous forme réseau vont correspondre à deux visions différentes du corpus des références téléchargées.

La première approche sera orientée « Fréquence de paires de modalités », chaque cellule de la matrice des modalités va contenir la fréquence de co-apparition des modalités dont elle est intersection. L'objectif est alors de répondre à des questions pouvant être :

- Si le champ retenu est le champ auteur « AU » : quels auteurs publient souvent ensemble.
- Si le champ retenu est le champ entreprise « OS » : quelles entreprises ont développé des programmes commun de recherche.
- …

La deuxième approche sera orientée « Nombre de modalités communes à deux références », chaque cellule de la matrice des références va contenir le nombre de modalités que les deux références dont elle est intersection contiennent en commun. L'objectif est alors de répondre à des questions pouvant être :

- Quels sujets de recherche appartiennent aux mêmes domaines scientifiques.
- …

Ces deux matrices, qui sont des expressions différentes d'une même réalité, peuvent être construites après application de filtres visant à réduire l'espace des modalités et/ou l'espace des références. Elles sont représentées ci-dessous :

	Modalité 1	…	Modalité j	…	Modalité n
Modalité 1					
…					
Modalité i			Fréquence de la paire (i,j)		
…					
Modalité n					

Figure 1-6 : Matrice symétrique des modalités.

	Référence 1	…	Référence j	…	Référence m
Référence 1					
…					
Référence i			Nombre de modalités que les références i et		
…			j ont en commun		
Référence m					

Figure 1-7 : Matrice symétrique des références.

La matrice des modalités est construite à partir du produit matriciel de la transposée de la matrice de présence-absence par cette même matrice alors que la matrice des références est à l'inverse construite à partir du produit de la matrice de présence-absence par sa transposée. Les matrices des modalités et des références sont symétriques.

Ces matrices peuvent alors être interprétées comme des matrices de proximité et subir d'autres traitements de nature bibliométrique : analyse des mots associés, co-citations, …

La figure ci-dessous illustre la facilité avec laquelle le lecteur sera à même de comprendre un phénomène représenté sous forme de réseau et ce comparativement à sa représentation matricielle.

L'information traduite sous forme de réseau est assimilée instinctivement et sans effort.

	Auteur 1	Auteur 2	Auteur 3	Auteur 4	Auteur 5
Auteur 1		3	0	1	1
Auteur 2	3		2	0	1
Auteur 3	0	2		1	1
Auteur 4	1	0	1		1
Auteur 5	1	1	1	1	

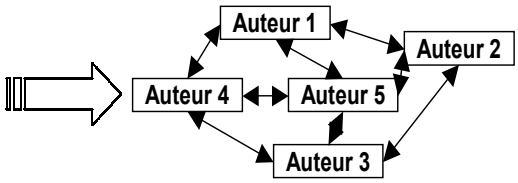


Figure 1-8 : Simplicité de la représentation réseau.

Il est ainsi aisé de conclure que l'auteur numéro cinq a une position centrale, et qu'il collabore avec l'ensemble des autres auteurs.

Dans l'exemple repris ci-dessous, l'interprétation de la matrice des références est aussi grandement facilité par l'expression réseau qui en est faite, il apparaît clairement que le corpus des références est scindé en deux sous-ensembles. Les références qui ont été extraites par la recherche bibliographique appartiennent éventuellement à deux thèmes de recherche distincts.

Il est néanmoins important de noter que si l'approche réseau permet de représenter la même information que l'approche matricielle mais avec moins d'éléments, celle-ci est encore améliorée par une disposition judicieuse des éléments qui la compose. Dans notre exemple le réseau est scindé en deux sous-réseaux tracés dans des espaces de représentation distincts.

	Réf. 1	Réf. 2	Réf. 3	Réf. 4	Réf. 5
Réf. 1		1	3	0	0
Réf. 2	1		1	0	0
Réf. 3	3	1		0	0
Réf. 4	0	0	0		2
Réf. 5	0	0	0	2	

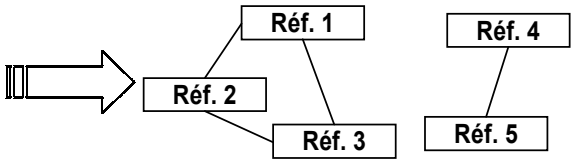


Figure 1-9 : Espace de tracé affecté à chaque sous-réseaux.

L'optimisation du positionnement spatial du réseau constitue l'objet principal de ce travail de recherche. Il s'agit ici d'organiser au mieux la représentation des matrices des références sous forme réseau, en vue d'en faciliter la compréhension.

Pour ce faire, aucun traitement matriciel n'a été implémenté, la démarche retenue a consisté à traiter la dimension géographique du problème à savoir le positionnement des points dans l'espace de tracé.

2 OPTIMISATION DE L'ESTHETIQUE D'UN GRAPHE

2.1 LISIBILITE D'UN GRAPHE

La construction d'un graphe comme nous venons de le voir s'appuie sur une structure matricielle qui pourra en bibliométrie être de type « modalités » ou « références ».

Néanmoins, une fois l'exploration de la matrice sous-jacente réalisée il va être nécessaire de positionner des sommets (modalités ou références) qui seront donc reliés par des arêtes (relations) dans l'espace de tracé. Cette opération n'est pas sans conséquence sur la lisibilité finale du réseau. Mais avant de détailler cette opération évoquons le terme lisibilité.

2.1.1 Lisibilité liée à la complexité du réseau

Le nombre de sommets mais aussi et surtout le nombre d'arrêtes vont être un facteur aggravant de la lisibilité d'un réseau. Il s'agit de la lisibilité intrinsèque du réseau. Ceci s'explique d'une part par la limitation de la capacité de traitement du cerveau humain et d'autre part par la limitation physique de l'espace de tracé. La forte concentration d'arêtes liant les individus crée une sensation de confusion.

La lisibilité intrinsèque peut être améliorée par l'application maîtrisée de filtres qui vont avoir pour effet de supprimer certaines arêtes n'autorisant par exemple que celles représentatives d'une fréquence de co-apparition des entités supérieure à un certain seuil. Seules les arêtes pertinentes seront alors tracées ce qui va par exemple mettre en évidence des sous-ensembles du réseau jusqu'alors masqués.

La lisibilité intrinsèque a fait l'objet de nombreux travaux au sein du CRRM et est notamment prise en charge par l'application MATRISME [Boutin-99]. Elle devra dans tous les cas être prise en considération, et ce avant de s'atteler à améliorer la lisibilité construite. En effet un réseau trop complexe ne pourra, même si le positionnement de ses sommets est optimisé, être représenté clairement dans un espace de tracé trop restreint au vu du nombre d'éléments qui le composent.

2.1.2 Lisibilité construite

Elle est illustrée par la figure ci-dessous. La lisibilité construite constitue l'objet de ce travail de recherche, il s'agit de mettre en évidence l'organisation entre les entités qui constituent le réseau et ce par des artifices de positionnement qui

s’appuient sur des critères d’esthétisme dont le plus évident est constitué par le nombre de croisements des arêtes du réseau.

Figure 2-1 : Critères d'esthétisme : croisements d'arêtes.

Ce réseau est constitué de deux sous-ensembles qui apparaissent plus clairement dans la représentation de droite.

2.1.3 Lisibilité contingente

Elle traduit le fait que chaque utilisateur va avoir tendance à modifier, même légèrement, la position de certains sommets du réseau de façon à obtenir une représentation qui lui convienne.

L’objectif de l’utilisateur étant soit d’améliorer sa perception du réseau, soit de mettre en évidence certains phénomènes pour lesquels il a de l’intérêt.

2.2 QUELQUES ALGORITHMES AUTOMATIQUES DE POSITIONNEMENT

Ci-dessous sont présentés quelques exemples d'applications relatives à notre problématique.

Il est important de noter que la grande majorité des travaux effectués dans le domaine de la simplification de l'esthétisme des graphes fait appel à la mise en œuvre de traitements stochastiques.

2.2.1 Heuristique de Eades

Le principe de cet algorithme est d'assimiler chaque arête d'un graphe à un élastique reliant des sommets.

Le graphe est initialement positionné de manière aléatoire dans l'espace de tracé. Puis le système évolue librement de manière à aboutir à un équilibre, où la somme des forces exercées par les élastiques est minimale.

Il est à noter que le traitement des forces répulsives est plus complet que celui des forces attractives appliquées par les élastiques, ce qui constitue une approximation dégradante.

2.2.2 Algorithme de Kamada et Kawai

Il s'agit d'une évolution de l'algorithme de Eades.

Cet algorithme intègre les interactions entre sommets non voisins, par le biais des plus courts chemins reliant ces sommets. Cet algorithme est basé sur la résolution d'équations différentielles et a notamment donné lieu à une application fournie avec un environnement de développement Java qui a été testé dans le cadre de ce travail.

Cet algorithme est très visuel et dynamique, il permet d'optimiser rapidement un certain nombre de conformations de graphes susceptibles d'être observées dans le «monde réel». Néanmoins sa mise en œuvre est délicate, elle implique une maîtrise fine de nombreux paramètres dont seule la « bonne » combinaison va influencer la convergence de l'algorithme.

2.2.3 Approche de Davidson et Harel

Il s'agit de l'application d'un algorithme de recuit simulé, l'objectif étant ici aussi de minimiser le niveau d'énergie globale d'un système.

Le niveau d'énergie est représentatif de la fonction d'esthétisme retenue qui intègre différents paramètres comme l'occupation de l'espace disponible de tracé, l'intersection des arêtes, la dimension des arêtes, …

2.2.4 La méthode de Fruchterman et Reingold

Là encore il s'agit d'une variante de l'heuristique de Eades.

Cet algorithme favorise la proximité des sommets liés par une arête, tout en les maintenant à une certaine distance minimale.

Il s'agit de la transposition informatique des observations faites en physique nucléaire. Les nucléons (sommets du graphe) sont soumis à des forces attractives qui tendent à les faire se rapprocher jusqu'à ce que la distance qui les sépare atteigne une valeur seuil à partir de laquelle une force répulsive devient prépondérante.

Au lancement cet algorithme va positionner les sommets des graphes de manière aléatoire. Il simulera ensuite l'application de forces physiques attractives qui vont avoir pour effet de concentrer les différents sommets du graphe dans un espace de tracé réduit. Des forces répulsives vont ensuite être simulées, celles-ci ayant des effets sur les déplacements des sommets qui iront en diminuant par la prise en compte d'un troisième paramètre : la température du système qui va en diminuant.

2.2.5 Algorithme génétique de Groves et Michalewicz

Il s'agit de l'application d'un algorithme génétique ayant pour objectif d'optimiser un graphe en prenant en compte un critère d'esthétisme paramétrable en fonction des objectifs à atteindre.

L'algorithme génétique retenu va prendre en charge la totalité de la problématique, la fonction d'adaptation étant calculée sur la totalité du graphe.

2.2.6 Synthèse

D'autres exemples peuvent être évoqués.

- L'optimisation des modèles conceptuels de Merise traitée par H. Heckenroth [Heckenroth-90] par application d'algorithmes de recuit simulé.
- Une approche plus mathématique, celle développée par M. Dalud [Dalud-94].

- L'application d'algorithmes génétiques pour le traitement de matrices de proximité traitée par I.C Lerman et R.F. Ngouenet [Lerman-94].
- …

Par contre la littérature ne recèle pas d'application combinant différents algorithmes d'optimisation stochastique.

Il en va de même pour le développement de chaînes de traitement hybridant les approches déterministes et probabilistes en vue d'en tirer le meilleur parti.

De plus, il ne semble pas possible d'aboutir à une méthode de traitement de l'optimisation des graphes s'appuyant exclusivement sur une approche déterministe, il semble que l'optimisation, notamment des composantes connexes des graphes, passe forcément par l'exploitation d'algorithmes stochastiques. La théorie des graphes n'étant pas encore assez avancée pour prendre en compte la totalité de cette problématique.

2.2.7 Apport de ce travail de recherche

Ce travail de recherche a pour objectif d'explorer les traitements susceptibles d'être mis en œuvre pour prendre en charge et de manière automatique l'optimisation de la lisibilité construite des réseaux.

L'orientation retenue est tout à fait innovante dans la mesure ou elle consiste à définir une chaîne de traitement hybride intégrant à la fois une dimension déterministe et stochastique.

La composante stochastique étant elle-même optimisée par combinaison de deux types d'algorithmes, un algorithme rapide de type recuit simulé, complété par un algorithme plus puissant mais moins dynamique à savoir une algorithme de type génétique.

Un paramétrage fin de cette chaîne de traitement va permettre de tirer le meilleur parti des différentes approches retenues.

2.3 UNE NECESSAIRE APPROPRIATION PAR L'UTILISATEUR

Comme cela a été expliqué précédemment l'exploitation qui va être faite d'un réseau est fonction de la nature de l'information recherchée ainsi que du décideur qui est en attente d'une information synthétique.

L'intervention finale de l'utilisateur dans la représentation du réseau reste toujours nécessaire, ne serait-ce que parce qu'elle favorise le phénomène d'appropriation du résultat de l'analyse.

C'est pourquoi il est important de prévoir, comme cela a été le cas pour le logiciel Matrisme développé par le Centre de Recherche Rétrospective de Marseille, une fonctionnalité autorisant la retouche manuelle du graphe obtenu suite à l'application d'algorithmes de positionnement automatique.

Les propriétés du réseau, susceptibles d'être ajustées par l'utilisateur, ont été mises en évidence par Freeman et Webster [Freeman-94], il s'agit de la forme ou de la couleur des sommets du réseau, de la couleur ou de l'épaisseur des arcs, de la position des sommets les uns par rapport aux autres ainsi que de l'exploitation d'animations permettant de visualiser une éventuelle évolution des informations traitées dans le temps.

Si une fonctionnalité de retouche manuelle du réseau doit être intégrée à tout logiciel de représentation de graphe, il est tout aussi important de prévoir une fonction d'annulation des dernières opérations effectuées. En effet plusieurs auteurs ont démontré qu'une modification mineure d'un des sommets du réseau peut changer complètement la perception de l'information contenue par celui-ci.

C'est ce qu'à notamment démontré McGraph [McGraph-97] par l'expérience illustrée ci-dessous, où le rôle d'isthme joué par les sommets O et B peut être masqué par le déplacement d'un seul sommet du réseau.

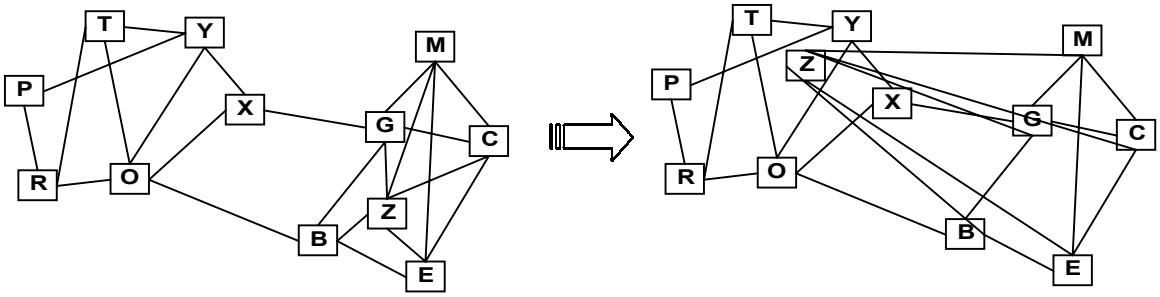


Figure 2-2 : Déplacement mineur du sommet "Z".

2.4 APPROCHE DETERMINISTE ISSUE DE LA THEORIE DES GRAPHES

2.4.1 Historique de la théorie des graphes

La théorie des graphes est le domaine des mathématiques initialement développé par Léonard Euler (1707-1783).

Le problème des ponts de Koenigsberg en est l'application la plus célèbre.

Euler modélisa les quartiers de cette ville sous la forme d'un graphe, ceux-ci étaient reliés entre eux par sept ponts (deux îles) enjambant la rivière Pregel.

Euler démontra que quel que soit le quartier de départ, on ne pouvait revenir à ce quartier en n'empruntant qu'une seule fois le même pont.

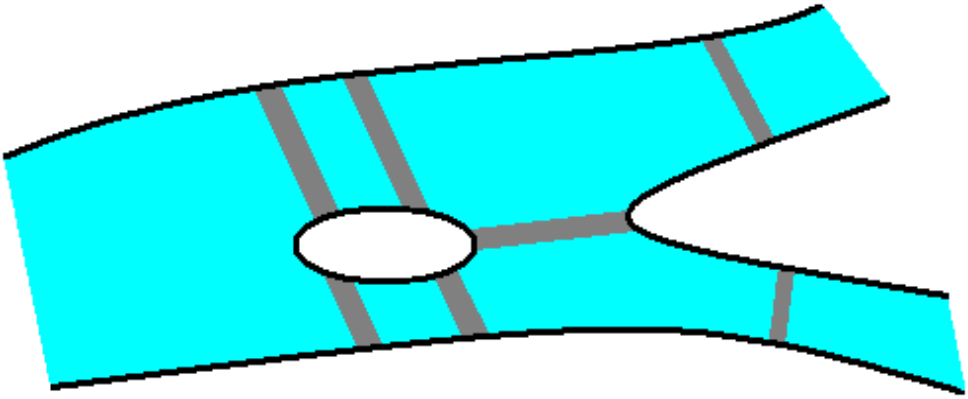


Figure 2-3 : Les ponts de koenigsberg.

Il généralisa ce problème à l'étude des graphes, cherchant notamment à répondre à la question suivante : dans quels graphes peut-on trouver un cycle [1] simple empruntant chaque arête une fois et une seule ? Il est à noter que les cycles de ce genre sont nommés cycles Eulériens.

La théorie des graphes était née, elle s'orienta ensuite vers la prise en compte de la notion de cheminement, les arêtes des graphes étant créées en fonctions d'objectifs à atteindre : optimisation des chemins [2], optimisation des flots, …

---

[1] C.F. § 2.4.4.11

[2] C.F. § 2.4.4.8

Il en va de même pour les cycles Hamiltoniens. Un cycle Hamiltonien est un cycle élémentaire qui passe une fois et une seule par chacun des sommets d'un graphe. En général ce type de cycle n'exploite pas toutes les arêtes d'un graphe, il n'en emprunte que deux par sommet.

Le problème du voyageur de commerce, qui est un problème de recherche opérationnelle, rappelle les cycles Hamiltoniens. Un voyageur cherche à visiter x villes puis à rentrer chez lui sans passer deux fois par la même ville et ce dans un temps minimal. Il n'existe pas de solution générale à ce problème, seuls des essais successifs sont susceptibles d'apporter une solution, quand celle-ci existe !

2.4.2 Quelques exemples d'applications de la théorie des graphes

2.4.2.1 Les couplages : le problème des mariages

Pour illustrer cette application, prenons pour exemple un groupe de garçons et un groupe de filles qui ont tous deux le même nombre de membres. Certains garçons connaissent déjà les filles et inversement. Résoudre le problème des mariages consiste à définir les conditions qui permettent d'apparier les membres de ces deux groupes de sorte que le garçon et la fille de chaque couple se connaissent déjà ?

2.4.2.2 Graphes orientés, plan de circulation, voies en sens unique

La question qui se pose alors est : peut-on conduire sa voiture d'un point quelconque de la ville à un autre en respectant les règles de circulation établies ?

2.4.2.3 La théorie des jeux (échecs, dames, …)

Les positions des pièces peuvent être assimilées aux sommets d'un graphe et leurs mouvements aux arêtes du graphe.

2.4.2.4 Le coloriage des cartes

Traité notamment par le mathématicien britannique Cayley. Les faces d'un graphe peuvent être assimilées à une carte. Dans un bon atlas, les divers pays et l'océan sont coloriés de diverses teintes qui permettent de les distinguer, le but étant d'obtenir deux régions ayant une frontière commune de couleurs différentes.

Le théorème des cinq couleurs précise qu'un graphe planaire [3] peut toujours être colorié avec seulement cinq couleurs différentes.

2.4.2.5 La théorie des réseaux de transport de Ford et Fulkerson

Elle est le fruit de recherches de certains spécialistes de la recherche opérationnelle et a donné lieu à de nombreux développements mathématiques notamment en programmation linéaire traitant d'optimisation combinatoire, de problèmes de cheminements et de circulation.

Ses applications interviennent dès qu'il s'agit de modéliser des systèmes techniques complexes formés d'une multitude de composantes élémentaires reliées les unes aux autres (réseau de distribution d'énergie, réseau routier, circulation d'information ou de pièces dans une entreprise, …)

2.4.3 Théorie des graphes et optimisation de la représentation graphique

Les quelques exemples cités plus haut sont représentatifs des orientations prises par la théories des graphes.

Les domaines d'applications moteurs ont été principalement la production industrielle et le commerce.

Il n'existe à ma connaissance pas d'algorithmes déterministes, découlant de la théorie des graphes et prenant en charge la simplification complète de leur représentation et plus précisément l'optimisation du nombre de croisements des arêtes, la prise en compte des poids des arêtes, des distances inter-sommets, ...

Ceci peut s'expliquer par les contraintes du monde réel : il faut minimiser un trajet, colorier une carte, réguler un flux de circulation ou encore optimiser une chaîne de production.

L'objectif de ce travail de recherche est différent, il s'agit de positionner un ensemble de sommets dans un espace de tracé, de manière à clarifier la structure des informations sous-jacentes au graphe.

Cet objectif implique donc la mise en œuvre d'algorithmes stochastiques.

2.4.4 Un référentiel pour mieux comprendre la structure des graphes

---

[3] C.F. § 2.4.4.20

Ci-dessous sont listées quelques définitions, notamment extraites de l'ouvrage de C. Berge [Berge-83], qui permettre de se familiariser avec la terminologie employée en théorie des graphes.

Cette liste de définitions permet non seulement de se familiariser avec les concepts manipulés en théorie des graphes, mais aussi, elle définit la terminologie qui sera exploitée dans la suite de ce document.

2.4.4.1 Graphe

L'équipe d'étude des graphes du Laboratoire Leibniz [www-leibniz.imag.fr-99] nous fourni la définition suivante : un graphe est une structure très simple puisqu'il est constitué d'un ensemble de sommets et d'une famille de liens (orientés ou non), appelés arêtes ou arcs, entre certains couples de sommets.

2.4.4.2 Arbre

Il s'agit d'un graphe particulier. En effet ici apparaît la notion de racine de l'arbre à laquelle les autres sommets sont rattachés et répartis de manière à former une partition en classes, chaque classe formant elle-même un arbre (sous-arbre).

Plus simplement, il s'agit d'un graphe connexe sans cycle.

La terminologie employée est celle des arbres généalogiques : sommet père, frère, fils …

Un arbre peut être binaire (deux nœuds fils liés à un nœud père) ou n-aire (jusqu'à n nœuds fils liés à un nœud père).

De nombreux manuels informatiques décrivent les algorithmes et structures de données appliqués au traitement des arbres. Nous retiendrons notamment les algorithmes d'exploration du type « Parcours symétrique », « Parcours en pré-ordre » et « Parcours Terminal » qui pourront être exploités dans l'analyse des graphes, ainsi que l'exploitation de matrices.

Il est important de retenir qu'un arbre à n sommets possède (n-1) arêtes, ce qui constitue un critère simple permettant de mettre en évidence une structure arborescente.

2.4.4.3 Arête

Courbe non orientée reliant deux sommets d'un graphe sans passer par aucun autre sommet.

2.4.4.4 Arête de cycle

Toute arête qui n'est pas un isthme.

2.4.4.5 Arête multiple

Si deux sommets d'un graphe sont reliés par plus d'une arête, chacune d'elles est dite arête multiple. Elles peuvent notamment traduire l'intensité d'une relation entre deux sommets.

2.4.4.6 Arc

Il s'agit d'une arête orientée.

L'école américaine assimile les graphes orientés à une particularité des graphes non orienté, ce qui est l'inverse pour l'école européenne.

2.4.4.7 Arête ou arc adjacents

Pour être adjacentes deux arêtes doivent avoir au minimum un sommet commun.

2.4.4.8 Chemin élémentaire du sommet i au sommet j

Il s'agit d'une succession d'arcs commençant au sommet i et se terminant au sommet j.

Le chemin est dit élémentaire si chacun des sommets n'est pas emprunté plus d'une fois.

Le chemin est dit simple si chacun des arcs n'est pas emprunté plus d'une fois. La longueur d'un chemin peut être égale au nombre de sommets parcourus ou à la somme des poids des arcs parcourus si le graphe est valué.

Un sommet seul est considéré comme un chemin de longueur 0.

2.4.4.9 Diamètre d'un graphe

Le diamètre d'un graphe est la longueur du plus long des chemins minimaux du graphe.

2.4.4.10 Chaîne

Séquence d'arêtes successives formant une courbe continue d'un sommet à un autre.

La chaîne est dite élémentaire si chacun des sommets n'est pas emprunté plus d'une fois.

La chaîne est dite simple si chacune des arêtes n'est pas empruntée plus d'une fois.

2.4.4.11 Cycle dans un graphe

Un cycle dans un graphe est un chemin se terminant sur le sommet de départ, chaque arête étant parcourue une seule fois.

Un arbre peut se définir comme un graphe acyclique.

Un cycle est dit élémentaire si chacun des sommets qui le composent n'est emprunté qu'une seule fois, à l'exception du sommet initial.

Un cycle est dit élémentaire ou simple s'il s'agit d'un cycle minimal (c'est-à-dire si on ne peut en déduire un autre cycle par suppression d'arcs).

2.4.4.12 Graphe connexe

Un graphe est dit connexe quand tout sommet de ce graphe est lié à chacun des autres sommets par une chaîne.

2.4.4.13 Connexité

Un graphe est connexe si pour chaque couple de sommets i et j il existe un chemin.

Un graphe peut être constitué de plusieurs composantes connexes.

La recherche des différentes composantes connexes pourra alors se faire par exploitation d'un algorithme de type « Depth First Search » notamment exploité pour parcourir les arbres n-aire.

Les traitements mis en œuvre ici sont peu coûteux en temps de calcul, ils sont en fait directement proportionnels aux nombre de sommets du graphe.

Cette remarque est particulièrement importance pour nous. Elle justifie à elle seule l'intérêt qu'il peut y avoir à exploiter une approche déterministe pour identifier les composantes connexes d'un graphe en vue de leur affecter un espace de tracé propre.

2.4.4.14 Graphe complet

Il s'agit d'un graphe comportant une arête joignant chaque couple de sommets, donc dont la connexité est maximale.

Si n est le nombre de sommets, le nombre d'arêtes d'un graphe complet peut être calculé comme étant égal à n(n-1)/2.

2.4.4.15 Graphe régulier

Il s'agit d'un graphe dont tous les sommets ont même degré (connectés au même nombre d'arêtes).

2.4.4.16 Clique

Une clique, d'un graphe G, est un sous-graphe complet de G.

2.4.4.17 Graphe valué

Il s'agit d'un graphe dont un poids (entier positif) a été affecté à chacun des arcs.

Il peut être représenté par une matrice carrée et symétrique de dimension n (nombre de sommets), dont M(i,j) = poids de l'arc reliant les sommets i et j ou zéro s'il n'y a aucun un arc reliant les sommets i et j.

2.4.4.18 Graphe orienté

Il s'agit d'un graphe dont chacun des arcs se voit attribué un sens. Il peut être représenté par une matrice carrée non symétrique.

2.4.4.19 Sous-graphe

Il s'agit d'un sous-ensemble du graphe initial.

2.4.4.20 Graphe planaire

Un graphe est planaire s'il peut être tracé dans un plan de façon que ses arêtes n'aient pas d'autres intersections ou points communs que les sommets (applications aux circuits imprimés). Cette définition a des implications importantes dans le cadre de ce travail.

2.4.4.21 Graphe isomorphe

Des graphes G1 et G2 sont dits isomorphes s'ils renferment la même information. Ils possèdent le même nombre de sommets et à tout couple de sommets du graphe G1 relié par une arête, (B1C1) par exemple, correspond un couple de sommets du graphe G2 relié par une arête (B2C2), et inversement.

2.4.4.22 Sous-graphe partiel d'un graphe G

Selon C. Berge [Berge-83], si un graphe G est le graphe des routes de France, la carte routière des routes nationales est un graphe partiel, la carte routière de la Seine-Maritime est un sous-graphe ; la carte routière des routes nationales de la Seine-Maritime est un sous-graphe partiel.

Cette analogie est très explicite.

2.4.4.23 Arbre de couverture minimal d'un graphe

C'est un sous-graphe à structure d'arbre contenant tous les sommets du graphe initial.

Si le graphe est valué, l'arbre minimal de couverture est un arbre de couverture de poids minimal.

2.4.4.24 Isthme ou pont

Une arête reliant un sommet A à un sommet B est qualifiée d'isthme si elle représente le seul moyen de passer de A à B et inversement.

Tout isthme divise les sommets d'un graphe en deux ensembles : ceux que l'on peut atteindre à partir de A sans traverser l'isthme et ceux que l'on peut atteindre à partir du sommet B sans traverser l'isthme, ce qui revient à partager le graphe en deux parties G1 et G2 qui ne sont connectées que par l'arête (A,B).

2.4.4.25 Point d'articulation

Un point d'articulation dans un graphe est un sommet dont la suppression augmente le nombre de composantes connexes, un isthme est une arête dont la suppression a le même effet. Nous verrons plus loin que cette définition est directement employée pour mettre en évidence les istmes et les points d'articulation et ainsi mieux appréhender la structure des graphes à traiter.

2.4.4.26 Corde

La corde d'un cycle élémentaire est une arête qui relie deux sommets non consécutifs de ce cycle.

2.4.4.27 Cactus

Un cactus est un graphe connexe dont chaque bloc est constitué soit par un isthme, soit par un cycle élémentaire sans cordes. Un cactus peut donc être représenté graphiquement par un ensemble d'arêtes sans intersections.

2.4.4.28 Nombre cyclomatique

Nombre d'arêtes du graphe diminué du nombre de sommets du graphe et augmenté de 1.

Il sera exploité comme critère de complexité d'un graphe.

2.4.4.29 Degré d'un sommet A

Le degré d'un sommet A est le nombre d'arêtes aboutissant à ce sommet A.

2.4.4.30 Sommet terminal

Il s'agit d'un sommet relié à une seule arête (son degré est de un).

2.4.4.31 Sommet isolé

Sommet auquel aucune arête ni aucun arc n'est incident.

2.4.4.32 Matrice d'adjacence

Il s'agit d'une matrice logique carrée et symétrique de dimension n (nombre de sommets) telle que : M(i,j) est vrai si les sommets i et j sont adjacents dans le graphe, c'est à dire liés par une arête.

Elle est construite en une seule opération par un recensement des nœuds adjacents appliqué à chacun des nœuds du graphe.

2.4.4.33 Liste d'adjacence

Il s'agit d'une série de lignes dont chacune d'elles correspond à une arête.

Visualisée sous forme de colonne, elle est composée de deux listes, la liste des sommets prédécesseurs (sommets liés au sommet en cours par un arc pointant vers le sommet en cours) et la liste des sommets successeurs (sommets liés au sommet en cours par un arc partant du sommet en cours).

Ces listes sont la base des recherches de chemin par algorithme d'exploration mis en œuvre dans en optimisation combinatoire.

2.4.4.34 Matrice d'incidence sommets-arcs

Il s'agit d'une matrice associée à un graphe ayant n sommets et m arcs.

Cette matrice comprend n lignes (correspondant aux sommets 1, …., *n*) et m colonnes (correspondant aux arcs u_1, …, u_m).

Chaque cellule a_{ik} de la matrice est calculée comme suit :

- +1, si le sommet *i* est extrémité terminale de l'arc u_k
- -1, si le sommet *i* est l'extrémité initiale de l'arc u_k
- 0 sinon

2.4.4.35 Matrice d'accès du graphe

Il s'agit d'une matrice logique carrée et symétrique de dimension n (nombre de sommets) telle que : M(i,j) est vrai s'il existe un chemin du sommet i vers le sommet j et M(i,j) est faux s'il n'existe pas de chemin de i vers j.

Elle peut être construite par application d'un algorithme de parcours en profondeur appliqué à chacun des sommets pour déterminer quels sommets on peut atteindre depuis celui-ci.

2.4.4.36 Matrice des distances

Il s'agit d'une matrice carrée et symétrique de dimension n (nombre de sommets) M(i,j) dont chaque cellule contient le nombre d'arcs constituant le plus court chemin du sommet i vers le sommet j, ou la valeur zéro s'il n'existe aucun chemin permettant de relier le sommet i au sommet j.

2.4.5 Exploitations de la théorie des graphes

2.4.5.1 Traitement de la connexité

Une première analyse de la structure du graphe peut se faire par application d'un algorithme du type « parcours en profondeur ».

Il s'agit d'un processus itératif qui consiste à explorer un graphe en traitant tous les sommets rencontrés jusqu'à aboutir à un sommet terminal ou un sommet dont tous les voisins ont déjà été explorés.

On rebrousse alors le chemin parcouru jusqu'à découvrir une nouvelle branche non encore explorée.

Le traitement prend fin quand l'ensemble des sommets du graphe a été exploré.

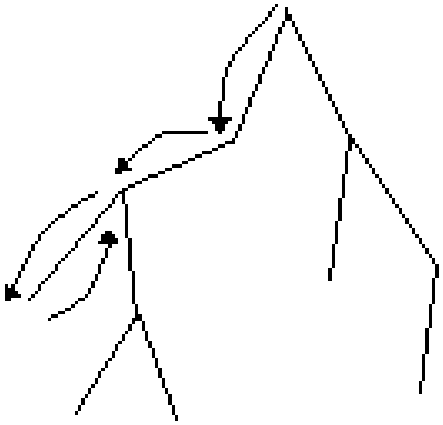


Figure 2-4 : Parcours d'un graphe de type arbre par une méthode dite "deep firth".

L'application de ce traitement va permettre d'obtenir un premier découpage du réseau en ses différentes composantes connexes.

Les sous-graphes pourront alors être positionnés dans des espaces de représentation disjoints, ce qui contribue à clarifier la représentation d'un graphe et donc déjà à améliorer la compréhension de l'information qu'il recèle.

De plus, l'optimisation prise en charge par d'autres traitements sera allégée.

Cette technique a été une des premières appliquées dans ce travail de recherche, l'exemple ci-dessous illustre la simplification apportée :

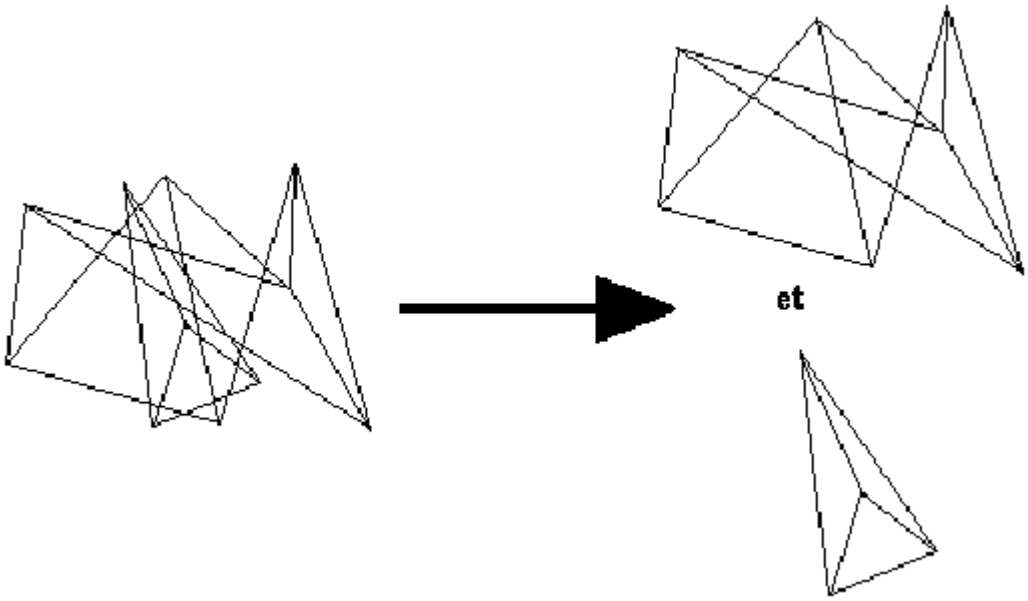


Figure 2-5 : Graphes et sous-graphes.

2.4.5.2 Recherche des composantes planaires

Comme nous l'avons vu plus haut un graphe planaire est un graphe qui peut être tracé dans un plan sans qu'aucune de ses arêtes ne se coupe en d'autres points que les sommets.

Cette propriété est très intéressante : en effet l'application de ce principe va permettre une analyse des composantes connexes à traiter, en vue de déterminer la possibilité de représenter un graphe sans croisement, ceci peut constituer la condition d'arrêt d'un algorithme d'optimisation.

Le mathématicien polonais Kuratowski a énoncé le théorème suivant : la condition nécessaire et suffisante pour qu'un graphe soit planaire est qu'il ne contienne aucun sous-graphe qui puisse être réduit à la forme pentagonale ou hexagonale ci-dessous dessinées :

Figure 2-6 : Planarité d'un graphe.

Sachant que réduire un graphe revient à remplacer, ses chaînes élémentaires sans arêtes latérales aux sommets intermédiaires, par de simples arêtes.

Exemple :

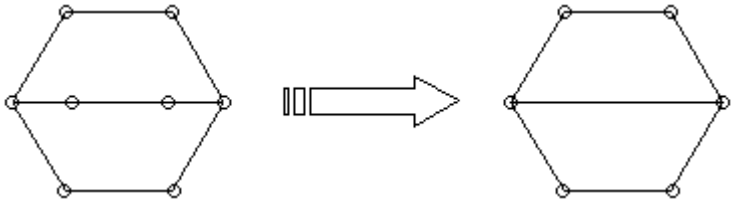


Figure 2-7 : Simplification d'un graphe par conservation des arêtes simples.

De nombreux algorithmes existent dans la littérature pour déterminer si un graphe est planaire ou non (Demoucron, Malgrange, Pertuiset, Lempel, Even, Cederbaum,…).
Le livre de Nishizeki et Chiba qui traite aussi de cette problématique peut aussi être « actuellement » consulté sur le site http://www-leibniz.imag.fr/GRAPH/francais/definitions.html.

2.4.5.3 Formule d'Euler

Elle concerne aussi les graphes planaires, mais plus particulièrement ceux formant un réseau polygonal dans le plan (carte des pays formant un continent).

Les frontières des pays peuvent être assimilées aux arêtes d'un graphe, les états constituent les polygones.

Euler a découvert la relation suivante : dans un graphe polygonal, si on considère qu'aucun graphe n'en contient un autre, n étant le nombre de sommets, m étant le nombre d'arêtes et f étant le nombre de faces, on peut alors écrire l'égalité suivante : $n - m + f = 2$.

Cette formule va nous procurer le moyen de déterminer si le graphe en cours de traitement forme un réseau polygonal et donc s'il est susceptible d'être représenté sans intersection d'arête.

La formule d'Euler peut donc aussi être exploitée comme condition d'arrêt d'un algorithme d'optimisation de tracé de graphes.

Il est à noter que dans le cas d'un graphe planaire le nombre cyclomatique correspond au nombre de faces du graphe.

2.4.5.4 Recherche des circuits minimums

De nombreux algorithmes sont disponibles dans la littérature, comme exemple l'algorithme de Bellman.

Celui-ci permet de calculer les longueurs des plus courts chemins dans un réseau orienté.

Cet algorithme offre aussi un intérêt pour les graphes non orientés dans la mesure où il est aisé d'orienter un graphe non orienté et ensuite d'appliquer cet algorithme pour ordonner les sommets du graphe avant application d'autres algorithmes d'optimisation de positionnement.

Des tests ont été effectués dans le cadre de ce travail. Mais cet algorithme n'a pas été intégré dans la chaîne de traitement définitive, en effet le complément d'informations apporté par cet algorithme n'a pas été valorisé par l'application des algorithmes stochastiques.

2.4.5.5 Dénombrement de chemins

Pour dénombrer, dans un graphe G = (X,U), les chemins entre chaque paire x et y on utilise le produit matriciel tel qu'il est défini en algèbre linéaire.

Si G est un graphe et A sa matrice associée, le coefficient p^i_j de la matrice $P = A^k$ (obtenu en effectuant k fois le produit de la matrice A par elle-même) est égal au nombre de chemins de longueur k allant de x_i à x_j dans le graphe.

De plus dans un graphe G, il existe un chemin de longueur k si et seulement si $A^K <> 0$.

Il n'existe pas de circuits si et seulement si $A^K = 0$ à partir d'un certain rang.

Ces énoncés peuvent servir de base à l'identification des composantes fortement connexes, ils ont été testés mais n'ont finalement pas été retenus dans le présent travail car ils se sont avérés trop consommateur de temps machine.

En effet, ils impliquent le maintien d'une structure de données complexe et de plus ils sont redondants avec les algorithmes d'identification des isthmes et des points d'articulation.

2.4.5.6 Détermination des isthmes et points d'articulation

Différents algorithmes sont disponibles dans la littérature.

Les plus simples découlent d'algorithmes de traitement de la connexité des graphes appliqués après suppression de sommets.

Ces algorithmes permettent de fractionner les graphes complexes et d'appliquer un pré-positionnement de ceux-ci.

La figure 2.8 illustre l'intérêt que peuvent avoir ces algorithmes.

L'augmentation des temps de traitement induite est faible, directement proportionnelle au nombre de sommets à traiter, c'est pourquoi il faut favoriser l'utilisation de cette approche.

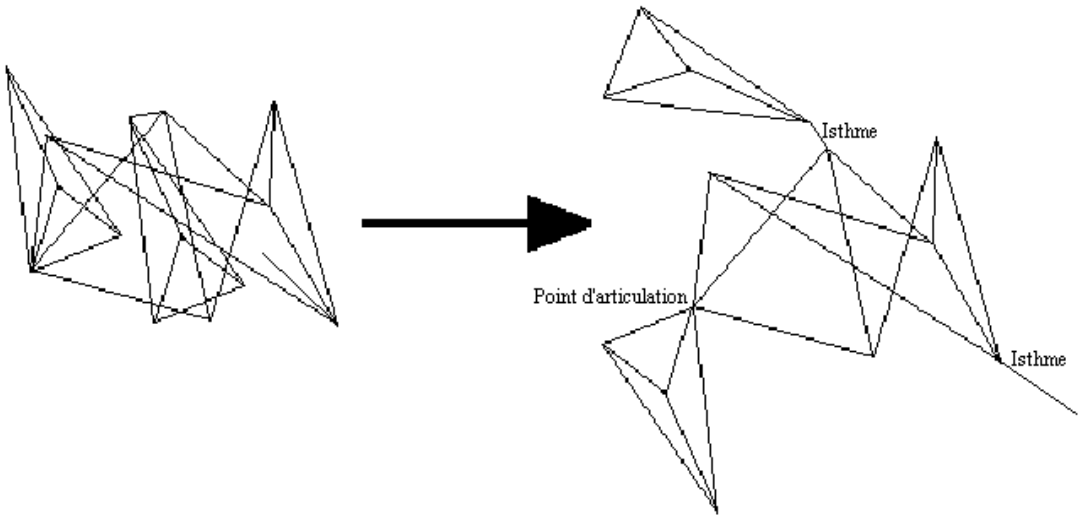


Figure 2-8 : Isthmes et points d'articulation dans un graphe.

2.4.5.7 Conclusion sur l'application de la théorie de graphes à notre problématique

Le traitement de la connexité va donc permettre d'explorer le graphe pour identifier les sous-graphes qui le composent, l'algorithme mis en œuvre étant relativement peut coûteux en temps de calcul.

La structure de ces sous-graphes pourra ensuite être analysée pour déterminer la présence d'isthmes et de points d'articulation.

Enfin la propriété de planarité de chacune des composantes connexes sera exploitée comme indicateur de fin de traitement.

2.5 APPROCHE PROBABILISTE : LE RECUIT SIMULE

2.5.1 Généralités

Le recuit est une méthode très employée en métallurgie, elle consiste à laisser refroidir un métal très lentement pour aboutir à une conformation moléculaire telle que l'énergie thermodynamique du système soit minimale.

Les molécules ont ainsi le temps et la possibilité de s'organiser en structures cristallines uniformes pour atteindre des états d'équilibre correspondant à des niveaux d'énergie thermodynamiques minimaux.

En 1953, Métropolis, travaillant sur la modélisation thermodynamique, a proposé une méthode de calcul des états d'équilibre thermodynamiques en milieu liquide.

Il s'agissait de simuler les mouvements de particules dans ce milieu et de calculer, pour une température donnée, l'énergie globale associée.

Considérons que chacune des coordonnées x et y des particules peut varier de Xmin à Xmax et Ymin à Ymax, selon les équations $x = x + a\,\Delta X$ et $y = y + b\,\Delta Y$, que les coefficients a et b appartiennent à l'intervalle [-1,1] et que la variation d'énergie engendrée par le déplacement d'une particule est notée ΔE.

L'algorithme créé par Métropolis implique que pour un ΔE positif, le déplacement aléatoire d'une particule est systématiquement accepté et pour un ΔE négatif, le déplacement de la particule a une probabilité d'être accepté si, P aléatoire et compris dans l'intervalle [0,1], est supérieur à $\exp^{(-\Delta E/kT)}$.

L'équilibre thermodynamique est atteint pour une température donnée avec la stabilisation de l'énergie du système. La recherche d'un équilibre est réitérée à différents paliers de température, celle-ci allant diminuant. L'énergie minimale du système est théoriquement atteinte à une température égale à 0°k.

Kirkpatrick [Kirkpatrick-83] a ensuite appliqué la méthode développée par Métropolis à des problèmes d'optimisation combinatoire.

Il a assimilé la configuration optimale d'un ensemble de particules à plusieurs degrés de liberté à une solution optimale pour un ensemble de paramètres combinatoires indépendants.

L'énergie du milieu étudié est alors assimilée au résultat d'une fonction à optimiser, la température du milieu étudiée étant aussi une caractéristique de la fonction à optimiser.

Le principe général est toujours de diminuer progressivement et par palier la température du système et de tendre pour chaque palier de température vers un état d'équilibre.

Ce travail a débouché sur une famille d'algorithmes nommée « Recuit Simulé ».

Ces algorithmes découlent donc directement de la thermodynamique.

Depuis leur création, leurs domaines d'application n'a cessé de croître, il s'étend de l'imagerie médicale [Alaoui-93] à la méthodologie informatique [Heckenroth-90], de la recherche opérationnelle à la thermodynamique ou encore au positionnement de circuits intégrés.

Cette méthode heuristique améliore réellement les temps de traitement observés avec les méthodes déterministes comme les méthodes de voisinage. Il suffit pour s'en convaincre d'évoquer le très célèbre exemple du voyageur de commerce qui est confronté à (n-1)!/2 solutions possibles s'il désire trouver le trajet le plus court lui permettant de visiter n villes sans passer deux fois pas la même (181440 combinaisons à évaluer pour seulement dix villes à visiter). Appliqué à ce problème, le recuit simulé est capable d'apporter une « solution acceptable » en évaluant un nombre de combinaisons très nettement inférieur à 181440 !

2.5.2 Algorithme général du recuit simulé

```
TempératureF = TempératureInitiale (température maximale retenue)
ConfigurationCourante = choix aléatoire d'une configuration initiale
F(ConfigurationCourante) = niveau d'énergie de la configuration courante
Répéter
  Répéter
    ConfigurationNouvelle = Génération aléatoire d'une solution voisine de la ConfigurationCourante
    Si f(ConfigurationNouvelle) < f(ConfigurationCourante) alors
      ConfigurationCourante = ConfigurationNouvelle
    Sinon
      Si (exp((f(ConfigurationNouvelle) – f(ConfigurationCourante)) / T) > ChoixAléatoire(0-1) ) alors
        ConfigurationCourante = ConfigurationNouvelle
      FinSi
    FinSi
  Jusqu'à ConditionEquilibreThermodynamique
TempératureF = TempératureF * FacteurDiminutionDeTempérature
```

Jusqu'à TempératureF = TempératureMinimaleDuSystème

Si nous retenons les abréviations suivantes :

Δf = (f(ConfigurationNouvelle) – f(ConfigurationCourante))

*T = Température*f

Exp(...) = exp(Δf/T)

Nous pouvons écrire que si Δf est inférieur à 0, alors Exp(...) appartiendra à l'intervalle [0-1].

Au lancement de l'algorithme, pour une température T importante, l'expression Exp(...) sera proche de 1 et la probabilité de retenir la « ConfigurationNouvelle » sera très importante, alors que celle-ci sera considérée comme ayant un niveau d'énergie supérieur à « ConfigurationCourante ».

Ceci constitue une dégradation du système que l'on cherche à optimiser, car si le système subit une amélioration il est systématiquement retenu, par contre si le système subit une dégradation, la probabilité de le retenir sera aussi relativement importante.

En fin d'exécution de l'algorithme, pour une température T faible, proche de 0, l'expression Exp(...) sera elle aussi proche de zéro et la « ConfigurationNouvelle » correspondant à une faible évolution de la configuration du système étudié aura une probabilité d'être rejetée beaucoup plus importante.

L'algorithme aura un comportement proche d'un algorithme classique de voisinage, ne retenant que les évolutions du système apportant une amélioration.

Pour des températures intermédiaires, on acceptera l'évolution du système avec une probabilité égale à Exp(...).

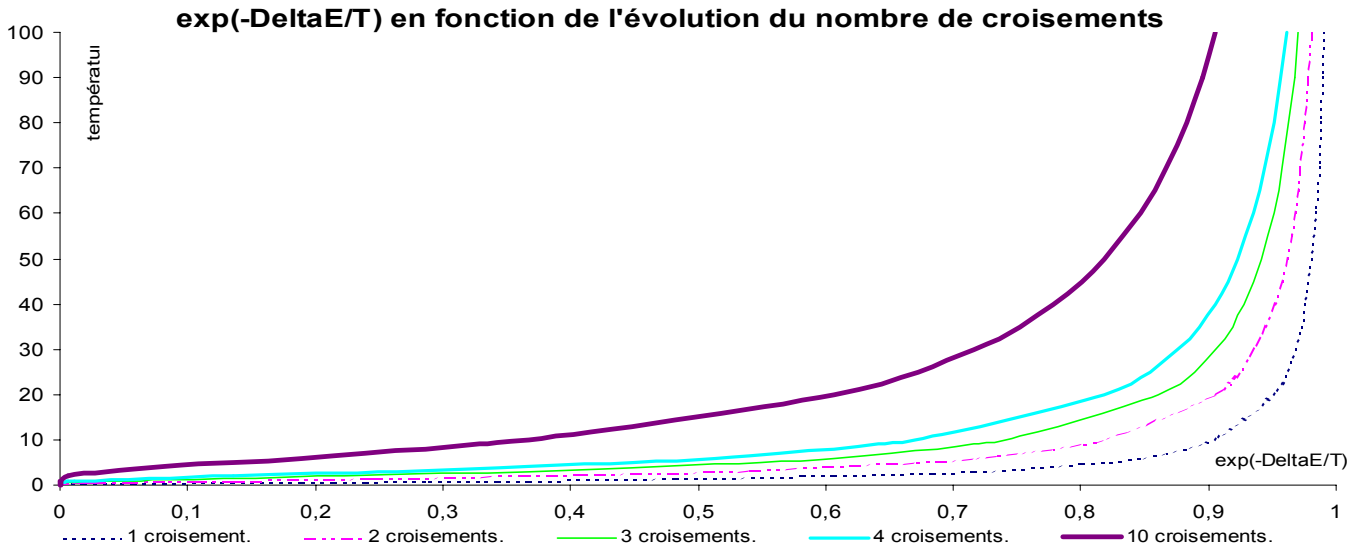


Figure 2-9 : appréciation du paramètre température

Cet algorithme constitue donc bien un heuristique, celui-ci ne nécessitant pas la mise en œuvre d'une condition de fin de traitement déterministe, condition que nous aurions bien du mal à définir dans ce contexte de recherche.

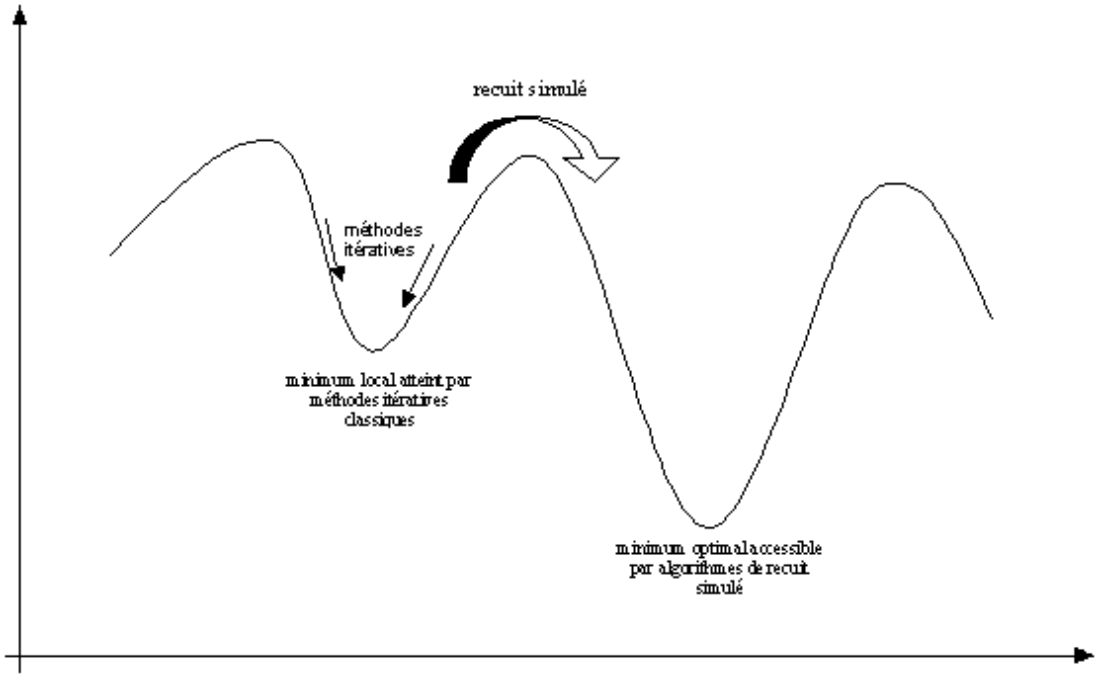


Figure 2-10 : Performance de l'algorithme de recuit simulé.

L'algorithme présenté plus haut est écrit en langage naturel, et bien qu'il s'agisse de l'algorithme général du recuit simulé, sa traduction directe en langage informatique aboutit à un code déjà opérationnel dont les caractéristiques sont celles citées plus haut.

Nous verrons plus loin quelles améliorations sont susceptibles d'être apportées dans le cadre de notre problématique.

Néanmoins il est important de signaler que pour un palier de température donné, nous n'allons pas tenter d'atteindre un état d'équilibre local. Par définition il s'agit d'une condition que nous ne pouvons évaluer. Nous allons simplement tester un nombre de conformations constant d'un palier de température à l'autre et mettre en œuvre un indicateur de progression de la fonction coût.

2.5.3 Application du recuit simulé à l'optimisation des graphes

Cet algorithme est tout à fait indiqué dans cadre de ce travail.

En effet, d'une part l'espace de recherche est très important et d'autre part il est tout à fait possible d'appliquer des changements élémentaires à un graphe, par déplacement « maîtrisé » d'un ou de plusieurs de ses sommets.

La fonction à optimiser est nommée fonction de coût, fonction objectif ou score.

Nous pouvons l'assimiler au critère d'esthétisme du graphe dont on cherche à optimiser la représentation.

Chaque coordonnée des points constituant le graphe est initialisée de manière aléatoire. Ensuite et à chaque itération, un ou plusieurs points sont déplacés aléatoirement dans l'espace de tracé affecté au graphe.

Au fil des itérations, la température baissant, la probabilité d'accepter un graphe ayant une conformation moins performante va diminuer. Au final, seules les améliorations de représentation du graphe seront retenues.

Le compromis performance / temps de traitement pourra être maîtrisé par les paramètres suivants de l'algorithme :

- La température initiale
- Le nombre de paliers de température (grandeur positive, qui va diminuant).
- Le nombre d'itérations à température constante.

Comme l'illustre la figure 2-9, il va être possible, en évaluant grossièrement le nombre de croisements supplémentaires créés ou supprimés par le déplacement d'un des sommets du graphe, d'appréhender l'ordre de grandeur de la température initiale, de la température finale, du nombre de palier à appliquer ainsi que la fonction d'évolution de celle-ci.

2.5.4 Fonction de coût

La fonction de coût retenue sera en fait la simple expression du critère d'esthétisme : le nombre de croisements calculé sur la base de la conformation du graphe courant. Le critère d'esthétisme retenu ne nécessite aucune codification, notre objectif est de minimiser sa valeur.

Il est important de dissocier le calcul de la fonction coût de l'algorithme général du recuit simulé, cela va nous permettre de faire évoluer le critère d'esthétisme retenu de manière indépendante à l'algorithme général d'optimisation des graphes.

Il est à noter que l'amélioration de l'organisation du graphe, apportée par l'identification des isthmes et points d'articulation, n'est pas dégradée par l'application de l'algorithme de recuit simulé.

L'ordonnancement des sommets retenu pour coder la première conformation du graphe à optimiser n'est pas modifié au fil du déroulement de l'algorithme de recuit simulé.

C'est aussi une justification de l'ordre d'application des algorithmes probabilistes dans la présente chaîne de traitement envisagée. Le recuit simulé va être exploité comme pré-traitement d'un algorithme génétique. Il va ainsi fournir un premier individu déjà « performant », qui sera ensuite exploité en entrée de traitement d'un algorithme génétique.

2.6 ALGORITHMES GENETIQUES

2.6.1 Généralités

Il s'agit d'une famille d'algorithmes qui tentent de traduire informatiquement la capacité qu'ont les organismes vivants à s'adapter à leur environnement et le fait que les individus les mieux adaptés ont une probabilité de survie plus importante et donc ont des caractéristiques qui seront globalement reproduites dans les générations suivantes.

Ces algorithmes s'inspirent de la théorie de l'évolution des espèces de Darwin, ils en reprennent d'ailleurs le vocabulaire.

Un parallèle peut être fait entre la « volonté » d'un organisme à s'adapter à son environnement, à optimiser ses caractéristiques de manière à mettre le maximum de chance de son côté pour survivre et la recherche de la conformation des paramètres d'une fonction correspondant à son optimum.

Pour expliciter ceci considérons une fonction à optimiser (recherche du maxima de la fonction) dont nous connaissons l'ensemble des paramètres.

L'objectif poursuivi est donc d'affecter une valeur à chacun de ces paramètres de telle sorte que la fonction à optimiser atteigne sa valeur maximale. Et ceci sans autre information sur la fonction à optimiser que le moyen de calculer la valeur de la fonction correspondant à une certaine combinaison de modalités de ses paramètres.

La première étape à mettre en œuvre pour aborder cette problématique à l'aide d'un algorithme génétique est d'adapter le vocabulaire utilisé plus haut.

La valeur de la fonction à optimiser peut être assimilée à l'indicateur permettant d'apprécier l'aptitude de l'espèce étudiée à survivre dans son environnement (critère de survie). Elle est nommée fonction d'adaptation.

Chaque jeu d'essai permettant d'initialiser les paramètres de la fonction peut être assimilé à un individu.

Plusieurs individus créés à un même instant T constituent une génération d'individus (chaque génération d'individus ayant le même nombre pair d'individus).

L'ensemble des générations créées pour rechercher la solution optimale peut être assimilé à la population totale des individus créés.

L'approche « algorithme génétique » va principalement consister à croiser entre eux, des individus appartenant tous à la même génération. Les individus les mieux « adaptés » à leur environnement auront le plus de chance de se reproduire pour constituer la génération suivante.

Chaque individu est constitué par la concaténation des valeurs prises par chacun des paramètres de la fonction d'adaptation à optimiser. Un individu est d'autant mieux adapté à son environnement que la valeur de la fonction d'adaptation à laquelle il correspond sera plus grande.

2.6.2 Algorithmes génétiques simplifiés

Ces algorithmes stochastiques permettent d'optimiser des fonctions ayant des espaces de paramètres continus ou discrets, ils appartiennent à la famille des méthodes de recherche basées sur l'exploration aléatoire.

Il n'est plus nécessaire de disposer d'informations complémentaires sur la fonction à étudier comme la continuité sur l'espace concerné, l'existence de dérivées, l'utilisation d'une expression analytique ou encore la résolution d'équations différentielles.

Seule la capacité à calculer la valeur d'une fonction à partir de l'ensemble de ses paramètres est nécessaire pour pouvoir les mettre en oeuvre.

Ceci explique que les problèmes non solvables par ces algorithmes sont peu fréquents.

De plus, il n'est pas utile de mémoriser l'ensemble des jeux d'essai mis en œuvre, la mémorisation de l'historique est intégrée dans l'algorithme.

C'est pourquoi dans de nombreux domaines ces algorithmes ont fourni des solutions à des problèmes jusqu'alors non résolus et se sont montrés plus robustes que les méthodes déterministes couramment appliquées jusqu'alors.

Ils traitent à chaque itération une génération d'individus et non un seul individu qu'ils tenteraient d'optimiser isolément. Ils explorent donc la totalité de l'espace de recherche traitant de nombreux points simultanément c'est pourquoi ils sont plus efficaces que d'autres algorithmes stochastiques (recuit simulé, …).

Ces algorithmes ont été créés par John Holland et son équipe à l'université du Michigan, son objectif était de créer des systèmes artificiels calquant les capacités d'adaptation des systèmes naturels.

Ils ont eu de nombreux domaines d'application notamment en optimisation de circuits de pipeline [Goldberg-91], en optimisation de sectorisation d'espaces aériens [Delahaye-95], en optimisation topologique [Kane-96], en robotique, en méthodologie informatique, …

Un exemple d'algorithme génétique simplifié est fourni par la figure 2.11. Il comporte plusieurs étapes :

- Codification de l'individu : l'ensemble des paramètres entrant dans le calcul de la fonction d'adaptation à optimiser est codé, si possible sous forme d'un digit, sinon sous la forme de plusieurs digits non sécables. Cet ensemble de codes est alors concaténé pour former une chaîne de caractères nommée individu.
- Fonction d'adaptation : pour chaque individu qui correspond à une conformation de paramètres, la fonction à optimiser sera calculée, elle restera liée à l'individu source du calcul et impactera sur sa capacité à participer à la création de la génération suivante. Cette fonction qui permettra d'évaluer la performance d'un individu intègrera dans le cadre ce travail une dimension esthétique.
- Opérateur de reproduction : il s'agit d'un opérateur probabiliste et non déterministe qui gèrera la capacité, la probabilité qu'aura un individu à intervenir dans la constitution de la génération suivante. C'est lui qui intègre le fait que plus un individu sera robuste (optimisation de sa fonction d'adaptation) plus il aura de chance de se reproduire.
- Opérateur de croisement : il s'agit d'un opérateur qui, à partir de certains individus appartenant à la génération précédente et sélectionnés par l'opérateur probabiliste de reproduction, va constituer la génération suivante. La nouvelle génération va être construite par le croisement de couples d'individus. C'est pourquoi chaque génération d'individu est constituée d'un nombre pair d'individus.
- Opérateur de mutation : pour maintenir l'analogie avec la génétique, on pourrait assimiler cet opérateur aux mutations génétiques engendrées par exemple par le rayonnement cosmique, … L'objectif poursuivi par cet opérateur est de limiter la perte d'informations intéressantes qui pourraient résider dans certaines parties de

graphes globalement non optimisés. Mais aussi d'explorer la totalité de l'espace de recherche.

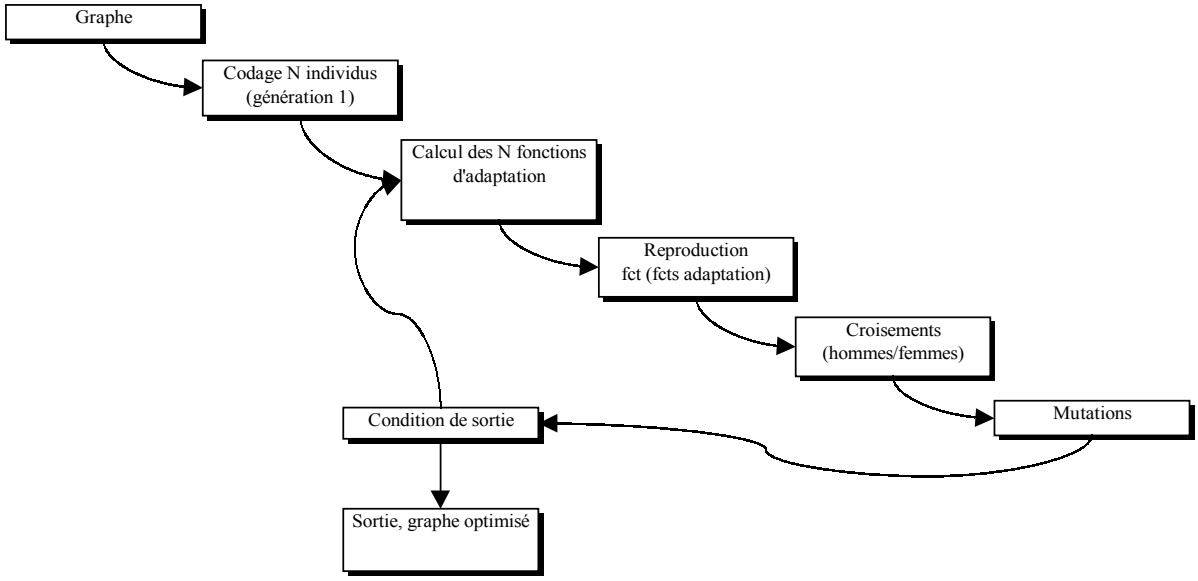


Figure 2-11 : Algorithme génétique simplifié.

2.6.3 Codification des individus

Les individus qui vont être traités par algorithmes génétiques sont ici les différentes conformations d'un même graphe.

Ces conformations d'un même graphe sont codées à l'aide de deux types (informatiques) de données :

- Les listes d'adjacence qui maintiennent l'information de type « arête ».
- Les coordonnées des sommets : un x et un y par sommet.

Seule les coordonnées des sommets vont être exploitées pour créer les individus.

Les listes d'adjacence seront exploitées pour calculer les fonctions d'adaptation des individus (ici fonction du nombre de croisements d'arêtes du graphe considéré).

Comme nous l'avons vu plus haut chaque génération est constituée par un ensemble pair d'individus.

Chaque individu est représenté informatiquement sous la forme d'une chaîne de caractères : concaténation des valeurs affectées à chacun des paramètres entrant dans le calcul de la fonction d'adaptation à optimiser.

La codification des coordonnées des sommets a un fort impact sur les temps de traitement.

Certains auteurs [Aubry-93] ont choisi de codifier leurs graphes dans un espace variable fonction de la dimension du graphe.

Dans le cadre de ce travail les individus seront codés dans un espace fixe fonction des caractéristiques du système d'affichage retenu ce qui permet d'éviter de ne traiter que des permutations de coordonnées tout en respectant le principe des alphabets minimaux.

Il est de plus important de ne pas perdre de vue que les dimensions de l'espace de recherche sont propres à chaque graphe traité.

Le développement d'un algorithme générique passe donc soit par l'exploitation d'un alphabet spécifique à chaque graphe, soit par un léger surdimensionnement de l'espace de recherche, solution retenue dans le cadre de ce travail.

Les coordonnées X et Y des sommets de chaque graphe seront concaténées pour former des chaînes de caractères de longueur fixe.

2.6.4 Génération initiale

Comme nous le verrons plus loin, les algorithmes génétiques sont coûteux en temps de traitement.

C'est pourquoi la première génération va découler d'un individu pré-traité par un algorithme de recuit simulé, plus rapide mais moins performant.

Cet algorithme va être à l'origine de l'individu « originel » qui, ayant subit des modifications mineures, va autoriser la création de la première génération.

2.6.5 Opérateurs de reproduction

Cet opérateur est, avec l'opérateur de croisement, à la base des algorithmes génétiques.

Considérons une génération d'individus. Chacun de ces individus est caractérisé par la valeur de la fonction d'adaptation qui lui est associée.

Si nous considérons que notre objectif est d'optimiser la fonction d'adaptation, nous recherchons donc à obtenir l'individu pour lequel la fonction d'adaptation est maximale.

Nous pouvons donc considérer que plus la valeur de la fonction d'adaptation associée à un individu est élevée, plus il sera probable que l'individu en question ait des caractéristiques proches de l'individu dont la fonction d'adaptation a la valeur maximale.

L'opérateur de reproduction tente de traduire ce concept.

Il permet de sélectionner les individus de la génération courante qui seront à l'origine d'une nouvelle génération.

La littérature fournit un grand nombre d'algorithmes permettant de coder l'opérateur de reproduction.

Ils peuvent se résumer à choisir de manière aléatoire les individus qui participeront à la création de la nouvelle génération parmi un ensemble d'individus constitué par des individus extraits de la génération courante avec une fréquence d'apparition proportionnelle à la valeur de leur fonction d'adaptation.

Il s'agit donc bien d'un opérateur probabiliste qui est de plus susceptible de retenir des individus qui sont globalement peu performants, mais qui peuvent néanmoins contenir une information qu'il pourrait être intéressant de maintenir dans la génération suivante, sous forme d'une sous-chaîne d'un nouvel individu.

Pour que la sélection aléatoire des individus soit la plus représentative de leur niveau de performance, il est donc important de maintenir des générations d'individus ayant une dimension minimale.

2.6.6 Opérateur de croisement

Cet opérateur permet, à partir des individus sélectionnés par l'opérateur de reproduction, de créer une nouvelle génération.

Le principe retenu dans le cadre des algorithmes génétiques simplifiés est très simple, il consiste à croiser deux à deux les chaînes constituant les individus sélectionnés et ce comme indiqué sur la figure 2.12.

L'analogie avec la biologie peut ici encore être faite, cet opérateur simule la combinaison du matériel génétique des parents qui intervient lors de la reproduction.

Il est toutefois à noter que si la nature exploitait réellement cet opérateur, les parents auraient systématiquement l'immense joie d'annoncer à leurs proches la naissance de faux jumeaux.

Chaque paire de chaîne est scindée au même niveau pour former des sous-chaînes qui seront ensuite interverties.

Cet opérateur est peu gourmant en terme de temps de calcul, la seule contrainte est que chaque génération soit constituée d'un nombre pair d'individus.

Il est de plus à signaler que pour les algorithmes génétiques simplifiés, la position de la césure de chacune des paires de chaîne qui constitue une génération, intervient de manière aléatoire.

Figure 2-12 : Opérateur de croisement en un point.

Les nouvelles chaînes générées sont différentes des chaînes parentes (dans la mesure où les deux chaînes parentes sont différentes) c'est pourquoi cet opérateur est fondamental dans le processus d'exploration. C'est un facteur de diversité.

2.6.7 Opérateur de mutation

Cet opérateur, contrairement aux opérateurs précédents, n'est pas systématiquement appliqué à chaque couple d'individus dans les algorithmes simplifiés.

Dans le domaine d'application qui nous intéresse, cet opérateur pourrait être assimilé à la capacité que l'on attribue à l'algorithme de créer, dans une génération nouvelle, un graphe qui globalement n'est pas optimisé, mais dont une partie seulement l'est.

Ou plus précisément, de créer un graphe globalement non optimisé, mais dont un sous-graphe pourrait l'être par exploration aléatoire et limitée de l'espace de variation d'un des sommets qui le constitue.

La figure 2.13 tente d'illustrer ce concept.

Les deux graphes représentés sont similaires, à l'exception de l'abscisse du sommet A. Celle-ci a été modifiée par l'opérateur de mutation.

Dans l'exemple présenté ci-après il se trouve que le graphe est optimisé par cette opération, ceci n'est pas le but recherché par l'opérateur de mutation qui n'est pas appliqué sous contrainte.

Le principal objectif recherché est d'éviter la perte d'informations engendrée par les autres opérateurs et qui pourrait être dommageable à la recherche de l'optimum global.

Nous avons vu plus haut que l'opérateur de croisement nous conduisait à « casser » des chaînes constituées d'une succession d'éléments indivisibles (les coordonnées X et Y des sommets des graphes).

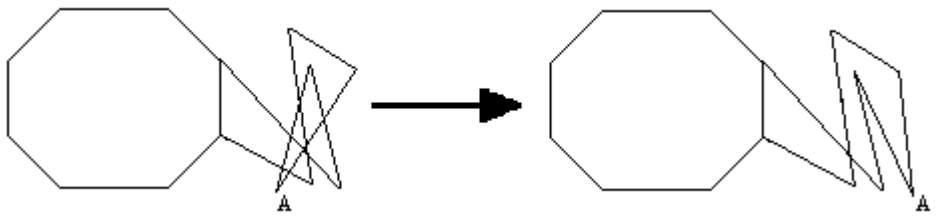


Figure 2-13 : Opérateur de mutation appliqué aléatoirement à la coordonnée x du sommet A.

L'opérateur de mutation intervient au niveau le plus élémentaire de la traduction informatique qui est faite de l'individu, au niveau des digits utilisés pour coder les chaînes de caractère.

En fait, il est appliqué avec une probabilité très faible. Dans la littérature il n'est pas rare de retrouver une fréquence proche d'un digit pour mille.

Cet opérateur est extrêmement coûteux en temps de calcul et ce à plusieurs titres :

- Il est appliqué avec une fréquence très faible, ce qui implique la gestion d'un système de compteurs lié au nombre de chaînes traitées. La fréquence d'application de l'opérateur est indépendante du nombre d'individus constituant une génération. Elle est globalement (choix aléatoire sous contrainte) fonction du nombre de

caractères traités et est donc susceptible d'intervenir de manière désynchronisée avec le cycle principal de l'algorithme génétique.

- Il traite des bits d'information et non des caractères eux-mêmes. L'appliquer consiste donc à appliquer l'opérateur booléen « NOT » avec, si la taille des caractères composant la chaîne est de huit digits, huit masques différents possibles.

Néanmoins cet opérateur doit être impérativement maintenu, la littérature est unanime.

2.6.8 Notion de schèmes

Les schèmes constituent une notion fondamentale dans les algorithmes génétiques, ils participent à en expliquer la puissance.

Pour mieux appréhender cette notion, revenons à la source des algorithmes génétiques : la biologie.

Le paragraphe 2.6.1 introduit le parallèle qui existe entre la génétique naturelle et les algorithmes qui en sont inspirés.

Le tableau ci-dessous [Goldberg-91] résume l'analogie entre les termes exploités dans les deux domaines.

Biologie	Algorithmes génétiques
Chromosome	Chaine de caractères
Gènes	Caractéristiques
Allèle	Valeur de la caractéristique
Locus	Position dans la chaîne
Génotype	Structure
Phénotype	Ensemble des paramètres
Epistasie	Non linéarité

Figure 2-14 : Correspondance de terminologie entre la biologie et les algorithmes génétiques.

Détaillons le.

Dans la nature, les caractéristiques propres à chaque individu sont « codées » dans son matériel génétique. Il s'agit de caractéristiques comme la taille, la couleur des cheveux, la couleur des yeux, le sexe, …, qui sont maintenues dans les chromosomes.

Un parallèle peut être fait entre les chromosomes et la structure de données autorisant l'enregistrement informatique de la combinaison des caractéristiques qui identifient un individu : la chaîne de caractères.

Chacune des caractéristiques dont la combinaison permet d'identifier un individu particulier, est maintenue, dans la nature, dans un gène différent, les algorithmes génétiques exploitent plutôt le terme de caractéristiques.

Une combinaison particulière de modalités des gènes, est nommée phénotype en génétique et ensemble des paramètres en informatique, il s'agit d'une solution particulière de l'espace de recherche.

Ces caractéristiques peuvent prendre des valeurs différentes d'un individu à l'autre, yeux noirs, yeux bleus, sensibilité au soleil ou encore résistance au paludisme. En génétique le terme allèle est exploité, sa traduction informatique étant « valeur de la caractéristique ».

En génétique le locus correspond à la position d'un gène dans un chromosome, sa traduction informatique est évidente : la position dans la chaîne de caractère codant un individu.

A un instant donné, la somme des caractéristiques de l'ensemble des individus vivants est nommée génotype en génétique, le terme structure est retenu en informatique.

Ces quelques définitions vont nous aider à mieux appréhender la notion de schèmes.

Dans la littérature, la notion de schème est souvent introduite à l'aide de l'exemple d'une fonction continue du type $f(x) = x^2$, optimisée par un algorithme génétique exploitant une représentation de x sous la forme binaire d'un entier codé sur un octet, variant de 0 à 255. Il est alors aisé de montrer que si les bits de poids fort prennent la valeur 1, la fonction d'adaptation aura une valeur plus élevée.

Nous venons d'appréhender de manière inconsciente la notion de schèmes, à savoir que les positions occupées dans la chaîne de caractères par les bits de poids fort vont avoir un impact plus important sur la valeur de la fonction d'adaptation (optimisation) que les bits de poids faible.

Nous aurions pu coder x, toujours sur huit bits, mais cette fois-ci sous la forme d'un entier court variant dans l'intervalle [-128,127]. Le bit de poids fort correspondant au bit de signe, les bits significatifs de l'optimisation de la fonction

d'adaptation auraient alors eu pour rang les positions 7, 6, 5, avec des valeurs initialisées à 1 ainsi que le bit 8 avec cette fois une valeur initialisée à 0.

Dans le cas de la fonction $f(x) = x^2$ ce sont les positions des bits les plus à droite qui sont concernées, dans la majorité des cas réels, la fonction d'adaptation à calculer est plus complexe, plusieurs ensembles de caractères constituant les individus peuvent avoir un impact particulier sur la fonction d'adaptation calculée.

Ceci est illustré à la figure 2-15.

Si la fonction d'adaptation est représentative de l'«adaptation» des individus à la vie dans des zones tropicales où le paludisme est endémique, les positions de rang 4 à 13 auront une influence toute particulière.

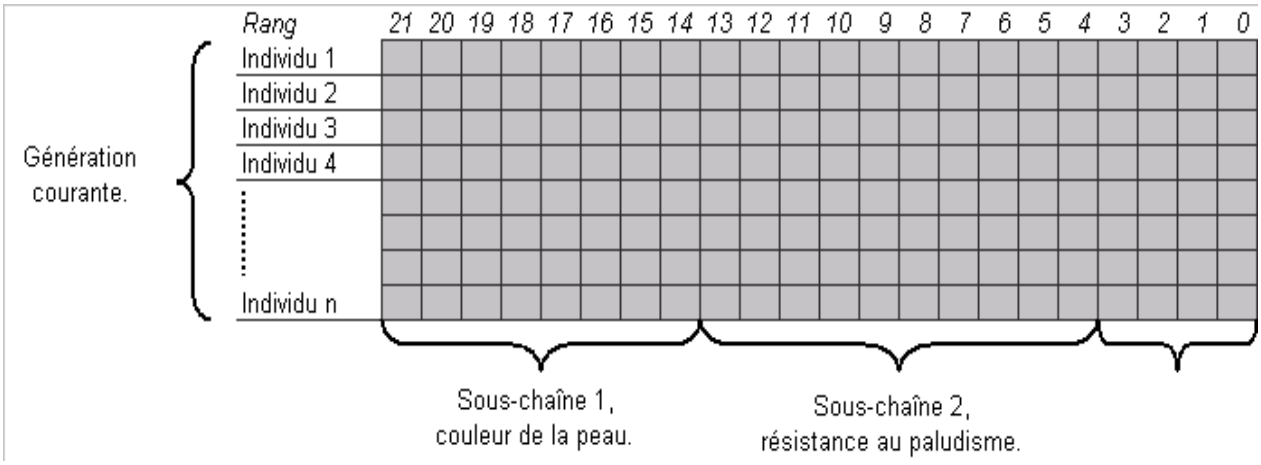


Figure 2-15 : Répartition des schèmes dans les chaînes codant les individus.

2.6.8.1 Théorème fondamental

La littérature abonde de démonstrations mathématiques explicitant l'influence des trois principaux opérateurs génétiques sur la reproduction des schèmes de génération en génération.

Ces démonstrations sont exploitables dans le cas de l'application stricte des trois opérateurs (reproduction, croisement et mutation) tels qu'ils sont définis plus haut.

Elles ne prennent pas en compte d'éventuelles améliorations des algorithmes génétiques dont certaines, nous allons le voir, sont incontournables.

Le théorème fondamental exprime le fait que les algorithmes génétiques ont tendance à favoriser de façon exponentielle la reproduction des schèmes courts dont la performance est au dessus de la moyenne de la population, mais détaillons le.

Nous avons vu que certaines positions de caractères dans la chaîne pouvaient avoir une influence particulière sur les caractéristiques de l'individu, engendrant une aptitude plus ou moins importante à se reproduire. Nous avons évoqué plus haut l'optimisation de la fonction $f(x) = x^2$, les individus étant constitués d'une seule variable codée sous forme d'entiers sur sept bits. Il s'agissait d'un exemple simple dans lequel un seul schème avait une influence importante, celui-ci étant constitué de positions (bits) regroupées en fin de chaîne. Cet exemple constituait une vision très simplifiée des structures d'individus mises en œuvre pour résoudre les problèmes d'optimisation de cas réels. En effet les algorithmes génétiques ont majoritairement à traiter des individus constitués de plusieurs variables concaténées pour former des chaînes de caractères, certains ensembles de positions dans les chaînes de caractères pouvant former des schèmes plus ou moins influents sur la capacité de reproduction des individus. Cette constatation nous amène donc à introduire deux propriétés relatives aux schèmes, à savoir l'ordre et la longueur utile.

L'ordre d'un schème correspond au nombre de positions occupées par celui-ci, que ces positions soient jointives ou non dans la chaîne de caractères exploitée pour coder l'individu.

La longueur utile du schème complète l'information fournie par la notion d'ordre, elle correspond à la distance entre le premier et le dernier caractère composant le schème.

Explicitons ces notions par un exemple et envisageons des individus codés à l'aide de chaînes de caractères de longueur l et dont chaque caractère pourra prendre une valeur notée 0 ou 1 si elles sont significatives et appartiennent à un schème ou bien encore une valeur notée * si elles n'appartiennent pas au schème considéré. L'ordre du schème compris dans la chaîne suivante « **1**001** » aura pour valeur 4, sa longueur utile étant de 6.

Nous avons aussi détaillé plus haut le principe de fonctionnement des opérateurs mis en œuvre dans le cadre des algorithmes génétiques simplifiés : les opérateurs de reproduction, de croisement et de mutation. Envisageons

maintenant leur effet sur la reproduction des schèmes de générations en générations [Goldberg-91].

L'opérateur de reproduction traduit le fait qu'un individu va participer à la constitution de la génération suivante en fonction de sa capacité à se reproduire, ou plus précisément en fonction de la valeur de sa fonction d'adaptation f_i. La probabilité p_i de participation d'un individu à la génération suivante est donc proportionnelle à sa fonction d'adaptation, elle est donnée par la formule $P_i=\frac{f_i}{\sum_{i=1}^{n} f_i}$. Si l'on note $m_{(H,t)}$ le nombre d'exemplaires d'un même schème H dans la génération de taille n à l'instant t, on pourra écrire qu'à l'instant t+1 il y aura $m_{(H,t+1)}$ exemplaires d'un même schème, $m_{(H,t+1)}$ étant égal à, $m_{(H,t+1)}=m_{(H,t)}*n*f_{(H)}/\sum_{i=1}^{n} f_i$ avec $f_{(H)}$ représentant l'adaptation moyenne des chaînes contenant le schème H à l'instant t. Si l'on note f l'adaptation moyenne de la population, celle-ci étant égale à $\sum_{i=1}^{n} f_i/n$, on obtient l'équation traduisant l'effet de l'opérateur de reproduction sur l'évolution des schèmes au fil des générations :

$$m(H,t+1)=m(H,t)*\frac{f(H)}{f}$$

Ce qui signifie que l'effet de l'opérateur de reproduction sur le nombre d'occurrences d'un schème H est fonction du rapport de son adaptation sur l'adaptation moyenne de la population. Les schèmes qui induisent des individus supérieurs à la moyennes vont donc se développer plus vite que les schèmes qui induisent des individus faibles. Considérons qu'un schème ait pour valeur de fonction d'adaptation $f(H)=c*f$, avec c étant une constante, nous pouvons réécrire l'équation précédente sous la forme suivante :

$$m(H,t+1)=m(H,t)*\frac{(f+cf)}{f}=(1+c)*m(H,t)$$

Cette équation peut aussi être écrite sous la forme de la suite géométrique suivante :

$$m(H,t)=m(H,0)*(1+c)^t$$

Les schèmes se reproduiront ou disparaîtront avec une vitesse exponentielle selon que leur fonction d'adaptation sera au dessus ou en dessous de l'adaptation moyenne des chaînes constituant une génération. Cette démonstration est à moduler par le fait que dans l'absolu l'adaptation moyenne d'une génération est susceptible d'évoluer au fil des générations. Néanmoins cette démonstration participe à expliquer la puissance de traitement des algorithmes génétiques.

Considérons maintenant l'opérateur de croisement, qui comme nous l'avons vu va permuter des chaînes appareillées, en s'appuyant sur un pivot dont la position est déterminée de façon aléatoire. Intuitivement nous pressentons que les schèmes de longueur utile importante vont être plus facilement détruits par cet opérateur. En effet plus la longueur utile $\delta(H)$ d'un schème sera importante, plus la probabilité de positionner le pivot de croisement dans le schème sera importante. Explicitons cette probabilité et notons p_d la probabilité qu'un pivot de croisement soit positionné dans un schème et donc la probabilité de destruction de ce schème. La probabilité de destruction d'un schème pourra donc s'écrire $P(d)=\frac{\delta(H)}{(l-1)}$. Le pivot de croisement pouvant se positionner sur les positions allant du deuxième caractère au dernier caractère de la chaîne la probabilité qu'un caractère appartenant à un schème soit atteint est proportionnel au rapport de sa longueur utile sur la longueur totale de la chaîne codant l'individu diminuée du premier caractère qui par définition ne peut être exploité comme pivot. La probabilité de survie d'un schème p_s pouvant être obtenue par l'équation $p_s = 1 - p_d$, nous pouvons alors écrire que :

$$p_s=1-\frac{\delta(H)}{(l-1)}$$

L'effet combiné des opérateurs de reproduction et de croisement est donc :

$$m(H,t+1)=m(H,t)*\frac{f(H)}{f^*[1-\frac{\delta(H)}{(l-1)}]}$$

La fréquence d'application de l'opérateur de mutation étant beaucoup plus faible que la fréquence d'application des opérateurs de reproduction et de croisement, son effet est négligeable sur la reproduction des schèmes de génération en génération.

L'application de l'équation précédente à notre sujet de recherche va se traduire par une observation extrêmement importante, à savoir la facilité qu'auront les algorithmes génétiques à optimiser les graphes constitués de plusieurs petits sous-graphes connexes, plutôt que les graphes composés de peu de gros sous-graphes connexes.

En effet, le principe de pertinence des briques élémentaires est respecté par le codage retenu.

Cette tendance pourra néanmoins être maîtrisée par l'application de certaines améliorations de l'algorithme génétique simplifié, comme la reproduction avec états stables.

2.6.8.2 Remarques

L'algorithme retenu ne tentera pas d'influencer l'évolution des schèmes au fil des générations.

Cela impliquerait une analyse en continue des sous-chaînes par des algorithmes déterministes en vue d'évaluer s'il existe des sous-structures de graphes déjà optimisées et ensuite de les repositionner dans les chaînes.

Une autre possibilité serait d'appliquer le calcul de la fonction d'adaptation à des portions de chaînes bien ciblées.

Dans les deux cas l'augmentation des temps de calcul est prohibitive.

Aucune de ces solutions n'a été retenue car par « nature » l'algorithme génétique intègre cette problématique.

Il en va de même pour l'exploitation de la dominance et de la diploïdie ou encore des opérateurs d'inversion qui pourtant mériteraient d'être retenus dans la chaîne de traitement s'ils n'étaient aussi gourmands en temps de calcul.

En effet considérer que les schèmes peuvent occuper des rangs différents d'une génération à l'autre ou même d'un individu à l'autre dans une même génération semble coller à notre réalité.

Comme le montre la figure 3-11, notre principal objectif et bien de déplacer les composantes connexes d'un graphe en vue d'améliorer un critère d'esthétisme.

Certains autres opérateurs de bas niveau n'ont pas non plus été intégrés dans la chaîne de traitement, toujours dans un souci d'optimisation des temps de calcul.

2.6.9 Améliorations des algorithmes génétiques

2.6.9.1 Maîtrise de la convergence des algorithmes génétiques

Nous avons détaillé plus haut le principe de sélection des individus, l'appréciation de la performance des individus étant assurée par le biais de l'opérateur de reproduction.

Il est précisé que cet opérateur est probabiliste, il ne s'agit pas d'une sélection systématique limitée aux individus les plus performants.

La caractéristique probabiliste de cet opérateur a été introduite dans le but essentiellement d'obtenir une sélection homogène, représentative des individus globalement les plus performants de la génération courante.

La sélection sera d'autant plus homogène que la dimension de chaque génération sera importante.

Il est clair que s'en tenir à cet opérateur peut conduire à une perte d'information, dans la mesure où certains individus globalement peu performants peuvent contenir des caractéristiques utiles à la recherche de la solution optimale.

C'est pourquoi l'opérateur de reproduction à été enrichi, D. Golberg [Golberg-91] nous fournit une liste d'algorithmes susceptibles d'être implémentés.

L'objectif poursuivi ici découle principalement du fait que l'application de l'opérateur de reproduction a tendance à favoriser la développement d'un individu très performants en début de traitement, puis à stagner autour de cet individu en fin de traitement.

Plusieurs solutions peuvent être adoptées pour palier cet inconvénient et améliorer l'espace des solutions exploré.

2.6.9.1.1 Transformation linéaire de la fonction d'adaptation

Une transformation linéaire peut être appliquée à la fonction d'adaptation : (a * (fonction d'adaptation) + b), l'objectif étant de réduire l'intervalle de variation de la fonction d'adaptation initiale.

Attention, il est important de veiller à ce que la moyenne de la fonction d'adaptation soit constante pour une même génération, ainsi un individu « moyen » restera moyen.

En réduisant ainsi l'intervalle de variation de la fonction d'adaptation, un super individu ne s'imposera pas en début de traitement, évitant ainsi une convergence prématurée de l'algorithme génétique, néfaste à l'exploration de la totalité de l'espace des solutions.

Il est courant, pour les dimensions de générations qui nous intéressent d'appliquer la règle suivante : fonction d'adaptation maximale recalculée (super individu) pour la génération en cours = 2 * moyenne des fonctions d'adaptation recalculées.

2.6.9.1.2 Windowing

L'adaptation de chaque individus est recalculée ainsi : elle sera égale à la différence entre son ancienne adaptation et le minimum des adaptations de la population totale jusqu'alors créée.

2.6.9.1.3 Troncature sigma

La nouvelle fonction d'adaptation est obtenue par la formule ci-dessous :

$$f' = f - (\overline{f} - c\sigma)$$

Où sigma est la variance des fonctions d'adaptations brutes de la génération en cours, et c une constante comprise entre 1 et 3.

2.6.9.2 Sélection des individus

Plusieurs auteurs ont pu mettre en évidence la supériorité d'autres opérateurs de sélection sur la très classique roue de loterie.

Comme exemple on peut citer la sélection stochastique pour la partie restante sans remplacement.

Il s'agit de calculer de manière classique la probabilité de reproduction de chacun des individus, puis de les sélectionner proportionnellement à la partie entière de leur probabilité de survie.

La partie décimale de leur probabilité de survie est ensuite exploitée par un tirage au sort biaisé en vue de compléter la liste des individus qui vont participer à la création de la nouvelle génération.

2.6.9.3 Ajustement de l'opérateur de croisement

L'opérateur de croisement a un effet destructeur sur les individus de la génération précédente.

En effet, hors mi le cas ou des individus identiques sont consécutivement sélectionnés par l'opérateur de reproduction, il n'est pas possible de maintenir des individus parfaits, ou du moins très performants d'une génération vers la suivante.

C’est pourquoi cet ajustement a été implémenté, il recopie donc systématiquement le(s) meilleur(s) individu(s) vers la génération suivante. Il améliore la convergence de l’algorithme tout en maintenant un parcours très complet de l’espace des solutions exploré.

2.6.9.4 Croisement uniforme

L’opérateur de croisement, présenté plus haut, implique un point de croisement unique de deux chaînes pères pour produire deux chaînes filles, il est à la base de la création d’une nouvelle génération.

Une évolution de cette opérateur consiste à croiser deux chaînes parentes en s’appuyant non pas sur un seul point pivot, mais en plusieurs points.

Cette évolution de l’opérateur de croisement va dans le sens d’un meilleur mélange du matériel génétique de deux individus et est plus conforme à ce qui est observé dans la nature.

Par contre non seulement il augmente les temps de calcul, mais de plus il accentue encore l’effet destructeur de cet opérateur sur les schèmes mis en évidence par les autres opérateurs appliqués dans le cadre de l’algorithme.

2.6.9.5 Reproduction avec état stable

Comme cela est indiqué plus haut l’opérateur de croisement crée à chaque génération un ensemble d’individus nouveaux, même si un ou plusieurs individus ont des caractéristiques exceptionnelles, ils seront dans tous les cas remplacés.

Pour pallier cette faiblesse un autre opérateur de construction de génération a été développé, il s’agit de la reproduction avec état stable.

Le principe de ce nouvel opérateur est simple, il consiste à supprimer les individus les plus mauvais de la génération en cours pour les remplacer par de nouveaux individus créés classiquement, pour aboutir à une nouvelle génération.

La nouvelle génération ainsi créée contiendra donc les meilleurs individus de la génération précédente, plus les meilleurs individus parmi les nouveaux créés, le ratio pouvant être maîtrisé par un taux de reproduction.

Le taux de reproduction avec état stable va permettre d’intervenir sur le compromis convergence - espace de recherche couvert, l’opérateur de mutation permettant de compenser une éventuelle diminution de l’espace de recherche.

2.6.9.6 Reproduction avec état stable sans duplication

Une variante de l'opérateur précédent consiste à supprimer ou éviter qu'il n'y ait trop d'individus performants identiques reproduits dans la génération suivante.

Ceci permet de mieux parcourir l'espace de recherche et d'introduire une certaine diversité dans la génération.

Le taux de duplication a aussi une influence sur le compromis convergence - espace de recherche couvert.

2.6.9.7 Complémentarité des opérateurs de croisement et de mutation

La maîtrise de l'influence conjointe des opérateurs de croisement et de mutation a été introduite dans les algorithmes génétiques.

L'objectif étant d'appliquer de manière aléatoire un de ces opérateurs à l'ensemble des individus de la génération courante, avec un taux d'application égal à 100%.

2.6.9.8 Adaptation dynamique de l'influence des opérateurs

Il peut être opportun d'adapter le taux d'application d'un opérateur en fonction de l'influence que celui-ci peut avoir sur l'optimisation globale obtenue.

Le principe est d'apprécier en continu la performance des opérateurs appliqués pour adapter leurs taux d'application.

Ceci permet améliore la robustesse des algorithmes génétiques, mais est néanmoins très coûteux en temps de calcul.

2.6.10 Hybridation des algorithmes génétiques

Les algorithmes génétiques sont essentiellement composés de trois opérateurs manipulant des individus dont l'objectif est de créer des générations contenant des individus de plus en plus « performants ».

Pour simplifier, nous pouvons considérer que ces opérateurs sont constitués par des d'entités de traitement distinctes qui sont appliquées de manière séquentielle dans un cycle.

Cette modularité explique le grand nombre de variantes et d'adaptions susceptibles d'être mises en œuvre pour optimiser une caractéristique bien

particulière, mais aussi la capacité qu'ont les algorithmes génétiques à s'hybrider avec d'autres algorithmes éventuellement orienté « métier ».

L'objectif poursuivi est alors d'optimiser l'outil de traitement dans le cadre d'une application particulière, on optimise alors la rapidité de traitements au détriment de leur robustesse.

Dans ce travail, cette orientation n'a été pas retenue, l'optimisation qui pouvait être prise en charge par la théorie des graphes a donné lieu à un traitement séparé, effectué en amont.

De plus le pré-traitement pris en charge par le recuit simulé se suffit à lui-même.

2.6.11 Optimiser une fonction

Comme nous l'avons vu plus haut, les domaines d'application des algorithmes génétiques sont vastes.

Ceci s'explique notamment par la simplicité mathématique des traitements mis en œuvre.

Comme cela a déjà été précisé, la connaissance des paramètres entrant dans le calcul de la fonction d'adaptation, ainsi que le moyen de la calculer sont suffisants pour optimiser une fonction.

La fonction à optimiser peut donc être quelconque, comme par exemple la réponse d'un autre algorithme de traitement exploitée en mode boîte noire.

De plus il est important de préciser le terme « optimiser » une fonction.

En effet, par une astuce mathématique, la valeur minimale du critère d'esthétisme retenu (nombre de croisements des arêtes d'un graphe) ici recherchée va être transformée en valeur maximale. Cette modification qui est une transformation linéaire intègrera aussi la notion d'intervalle de variation de la fonction d'adaptation allant de l'individu le plus performant à l'individu le moins performant.

2.6.12 Algorithmes génétiques appliqués à l'optimisation des graphes

Chaque conformation du graphe à optimiser sera donc codée sous forme d'un individu.

Chaque conformation est caractérisée par des parties plus lisibles (peu de croisements) ainsi que par des parties moins lisibles (nombre de croisements importants).

La figure 2-16 illustre le fait que le croisement aléatoire de deux graphes peut conduire à un graphe plus lisible.

Cette seule observation suffit à justifier la capacité qu'a un algorithme génétique à produire des graphes optimisés et ceci principalement pour deux raisons :

- Si l'on considère que l'on applique une astuce mathématique à la fonction d'adaptation d'un graphe pour convertir sont minima en maxima. Un graphe qui va contenir une forte proportion d'arêtes non croisées aura une valeur de fonction d'adaptation importante et aura donc une probabilité accrue d'être retenu pour former un nouveau graphe.
- Le nouveau graphe ainsi formé aura à son tour une plus forte probabilité d'être retenu pour engendrer de nouveaux graphes. Les nouveaux graphes ainsi formés auront une probabilité plus importante d'hériter de parties de graphes déjà optimisées.

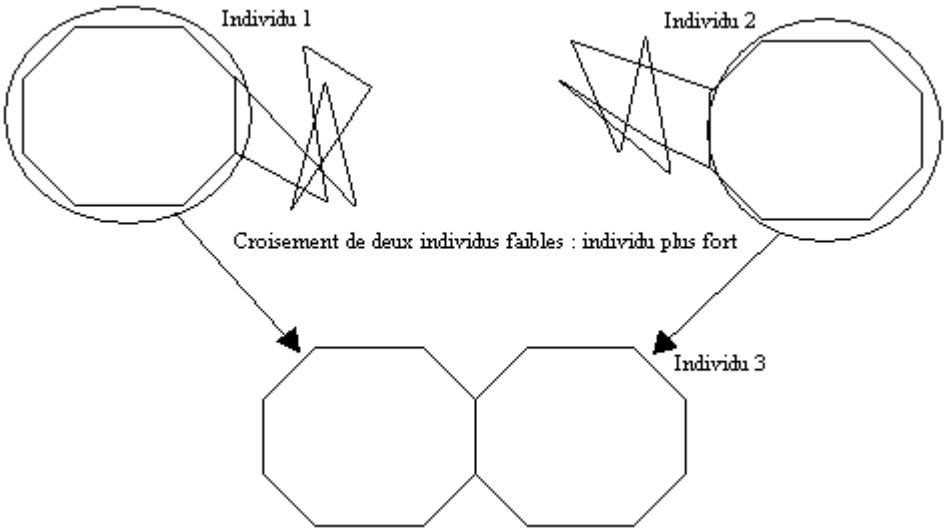


Figure 2-16 : Croisement favorable de deux individus "faibles".

Nous pouvons remarquer que les individus dont le nombre de croisements est plus faible auront une probabilité de se reproduire plus importante. De ce fait ils seront susceptibles d'être accouplés à d'autres individus eux aussi optimisés.

Le déroulement de cet algorithme va donc produire des générations de graphes dont le nombre de croisements va avoir tendance à diminuer.

C'est pourquoi les algorithmes génétiques sont capables de fournir des graphes optimisés du point de vu du nombre de croisements et donc de la clarté de lecture.

2.6.13 Conclusions sur les algorithmes génétiques

Ces algorithmes seront à même de résoudre les problèmes d'organisation des graphes ayant une forte connexité, problématique non prise en charge par des algorithmes déterministes.

Ils sont extrêmement puissants et robustes, ce qui explique le récent engouement qu'ils suscitent et le nombre croissant d'applications qui en est fait.

Par contre, ils ont comme inconvénient une consommation importante de temps machine. C'est pourquoi nombre de leurs applications font appel à des calculateurs massivement parallèles.

En effet, chaque génération d'individus créée peut être considérée comme un vecteur de fonctions d'adaptation (le plus souvent des polynômes) qui sont indépendants les uns des autres.

Les calculs à mettre en œuvre pour chacune des générations ne comprennent aucune boucle de traitement ou autres inhibiteurs de traitement parallèle.

L'ensemble des fonctions d'adaptation d'une génération peut donc être traité simultanément par un processeur parallèle.

A l'occasion de cette recherche nous n'avons pas envisagé ce type d'environnement informatique.

D'une part nous ne disposions pas de ce type de calculateur et d'autre part la chaîne de traitement envisagée est destinée à améliorer la lisibilité et donc la compréhension des graphes visualisés sur des unités légères de traitement (ordinateurs personnels).

C'est pourquoi il a été décidé d'abandonner certains opérateurs génétiques de bas niveau (opérateur d'inversion, …) et de coupler un algorithme génétique à un algorithme moins gourmant en terme de charge CPU : le recuit simulé.

L'algorithme de recuit simulé intervenant en pré-traitement d'un algorithme génétique.

Ces deux algorithmes étant enchaînés pour traiter les parties fortement connexes des graphes, comme indiqué sur la figure 2-17.

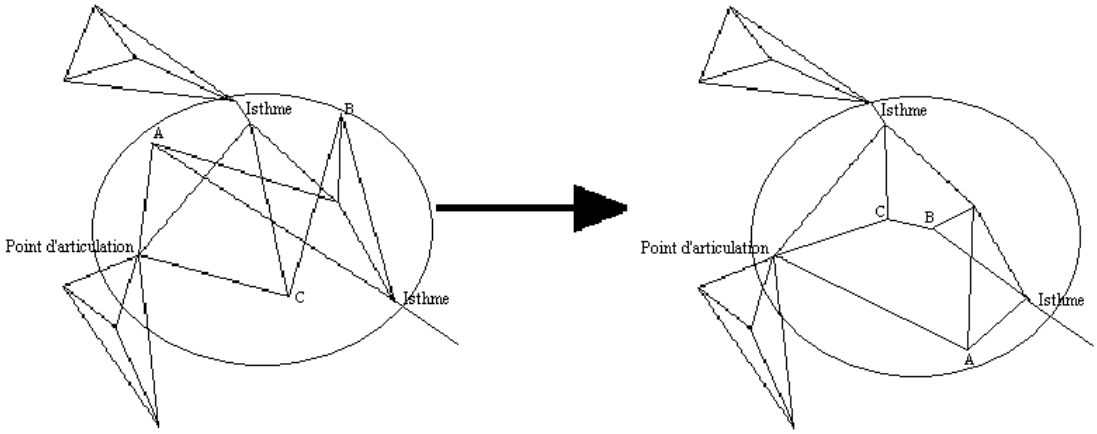


Figure 2-17 : Traitement de la partie fortement connexe d'un graphe par des algorithmes stochastiques.

2.7 SYNTHESE DES APPROCHES DETAILLEES PRECEDEMMENT

La théorie des graphes nous fournit donc un ensemble d'algorithmes déterministes robustes qui permettent de structurer les graphes à traiter en sous-graphes, en isthmes et en points d'articulation comme cela est détaillé en annexe 1. Ces algorithmes vont, de plus, permettre de déterminer si un graphe est planaire ou non, ce qui constitue une condition de fin de traitement d'autres méthodes d'optimisation.

L'algorithme de recuit simulé constitue une approche stochastique très utilisée en optimisation de structure. Il s'agit d'un algorithme relativement peu gourmand en temps de calcul, susceptible de prendre en compte de manière globale l'ensemble de la problématique. Il permet d'optimiser une fonction en évitant de s'enfermer dans un minimum local. Ici il sera utilisé pour pré-traiter les sous-graphes identifiés par les algorithmes déterministes.

L'algorithme génétique constitue lui aussi une approche probabiliste. Il est plus robuste que l'algorithme de recuit simulé mais aussi plus coûteux en temps de calcul. Il sera exploité en fin de chaîne de traitement, l'algorithme de recuit simulé ayant déjà commencé à organiser chaque composante connexe du graphe à optimiser, l'algorithme génétique va en affiner le traitement.

2.8 CHAINE DE TRAITEMENT ENVISAGEE

2.8.1 Détermination et séparation des sous-graphes

Par application d'algorithmes issus de la théorie des graphes.

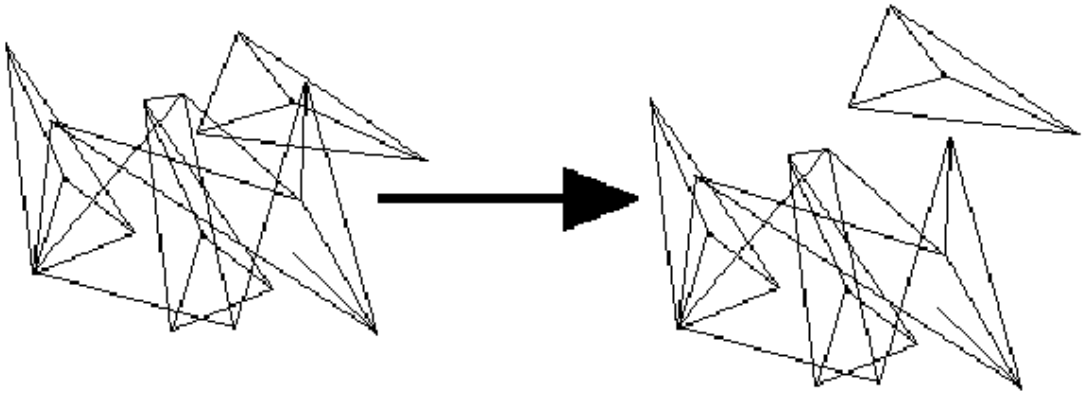


Figure 2-18 : Séparation du graphe principal en sous-graphes.

2.8.2 Identification des isthmes, points d'articulation et pré-positionnement

Par création d'algorithmes déterministes issus aussi de la théorie des graphes.

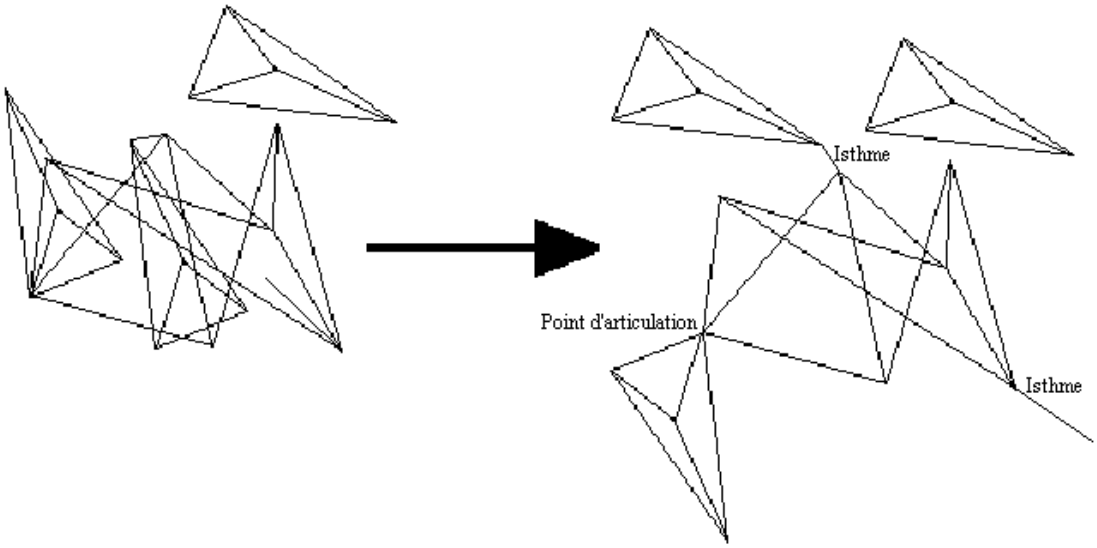


Figure 2-19 : Pré-positionnement des composantes connexes liées par des isthmes et points d'articulation.

2.8.3 Optimisation des croisements d'arcs dans les graphes connexes

Par application d'un algorithme de recuit simulé et d'un algorithme génétique adapté à la problématique « Optimisation des graphes ».

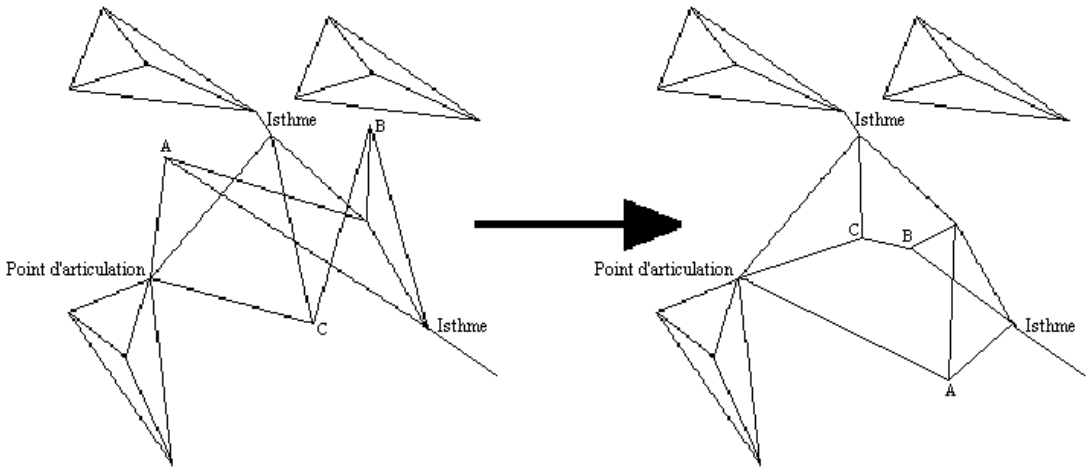


Figure 2-20 : Optimisation des parties connexes par approche stochastique.

2.8.4 Conclusions

L'objectif est donc d'améliorer la compréhension de l'information représentée sous forme de graphes, par application d'une chaîne de traitement générique, susceptible de traiter des graphes de volume important à l'aide d'unités de traitement légères (ordinateurs personnels).

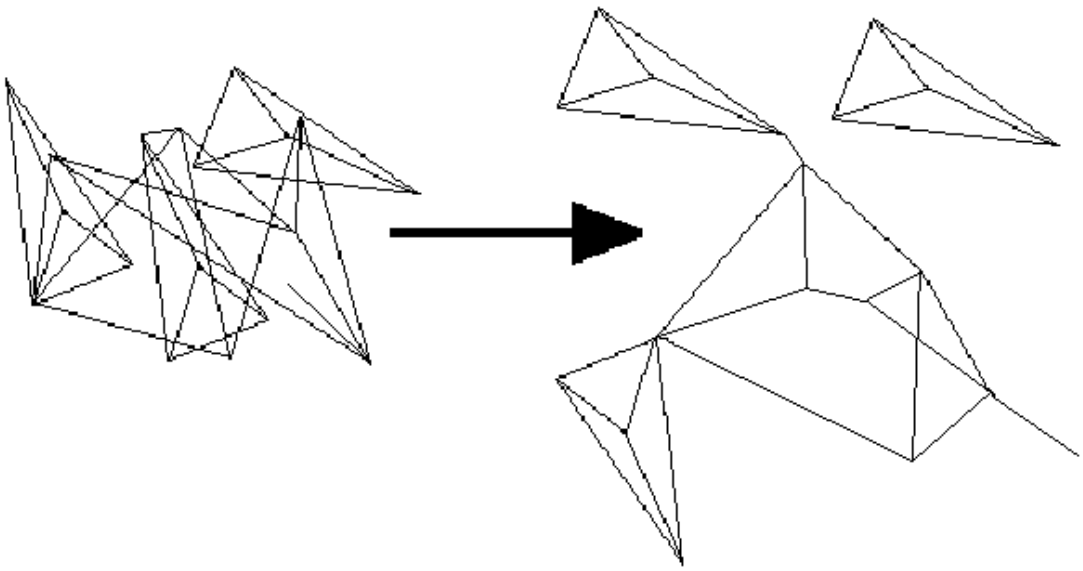


Figure 2-21 : Problématique prise en charge par la chaîne de traitement.

3 TEST DES ALGORITHMES

Un logiciel spécifique a été développé pour répondre à deux objectifs :

- Orienter et valider les choix théoriques effectués.
- Ajuster les paramètres impliqués dans les différents algorithmes mis en œuvre.

Il ne sera pas exploité tel quel en « production », mais les algorithmes qu'il aura permis de créer et de mettre au point seront intégrés dans le logiciel « MATRISME » [Boutin-99] qui intègre la grande majorité des fonctionnalités utiles au traitement des graphes.

Les principales fonctionnalités de l'application de mise au point sont :

- Chargement et codage d'un graphe.
- Sélection des algorithmes à appliquer.
- Sélection d'une fonction de coût.
- Paramétrage des différents algorithmes appliqués.
- Enregistrement et affichage des simulations.
- Export d'informations relatives à l'évolution de la fonction de coût vers un tableur et génération d'un rapport.
- Optimisation des graphes.
- Affichage et manipulation des graphes.

La forme principale de l'applicatif d'expérimentation est fournie à titre d'exemple en annexe 2, l'environnement d'exploitation retenu étant Windows.

3.1 BASE DE DONNEES DE GRAPHES A OPTIMISER

La mise au point de la chaîne de traitement s'est appuyée sur des graphes théoriques ayant des caractéristiques simples permettant de mieux en apprécier la performance.

L'objectif visé étant de juger la capacité de la chaîne de traitement à :

- Scinder le graphe initial en sous-graphes et les répartir dans des espaces de représentation distincts.
- Analyser chaque sous-graphe et identifier les composantes élémentaires (cf. annexe 1).
- Identifier les isthmes.
- Identifier les points d'articulation.
- Optimiser les composantes connexes de chacun des sous-graphes.

Des graphes réels filtrés ont été initialement exploités.

Les différents essais effectués ont permis de conclure que même si les graphes réels produits par exemple en bibliométrie sont plus complexes, ils n'apportent que peu d'intérêt comme base d'expérimentation.

En effet les graphes réels bruts ne sont jamais exploités tels quels, mais doivent préalablement être retravaillés avec des produits comme « MATRISME » [Boutin-99]. Ces produits disposent notamment de fonctionnalités permettant d'appliquer un certain nombre de filtres à des graphes, de manière à ne retenir que les informations stratégiques.

Ces opérations tendent à rapprocher les conformations des graphes réels et des graphes théoriques avec mise en évidences de sous-graphes, de sous-structures fortement connexes, d'isthmes, de points d'articulation, …

Un panel de graphes théoriques a donc finalement été retenu, ils sont représentatifs des conformations rencontrées dans les graphes réels retravaillés manuellement.

Les critères retenus pour construire la base de test ont été les suivants :

- Nombre de sommets.
- Nombre de sous-graphes.
- Nombre d'isthmes et répartition.

- Nombre de points d'articulation et répartition.
- Nombre de croisements d'arêtes.

L'application initiale d'un algorithme déterministe qui tend à découper le graphe en sous-graphes disjoints et à les représenter dans des espaces de tracé différents nous permet d'envisager une base de test plus restreinte.

Il n'est pas nécessaire de tester l'efficacité de la chaîne de traitement sur un panel de graphes combinant des sous-graphes ayant ou non des caractéristiques différentes, en effet ceux-ci seront optimisés séparément par les algorithmes probabilistes.

Les graphes retenus pour tester la chaîne de traitement sont détaillés au paragraphe suivant, ils sont représentés sous une forme déjà optimisée qui correspond à la cible à atteindre.

3.1.1 Caractéristiques des graphes de référence retenus

Ces graphes doivent permettre d'apprécier la capacité de la chaîne de traitement à optimiser des graphes ayant des conformations caractéristiques des critères définis au paragraphe précédent. Il faut tenter de couvrir tous les cas susceptibles de se présenter dans la réalité.

L'analyse des algorithmes appliqués permet d'affirmer que pour un paramétrage donné de la chaîne de traitement, le temps de calcul lié à l'optimisation du graphe est indépendant du nombre d'arêtes.

Seul le calcul du critère esthétique est fonction du nombre d'arêtes (évaluation de la fonction d'adaptation).

C'est pourquoi des graphes ayant un nombre constant d'arêtes ont été retenus (ce nombre a été fixé arbitrairement à 25).

Ceci permet de s'affranchir des temps de calcul de la fonction d'adaptation et d'observer des temps de traitement représentatifs de l'algorithme d'optimisation.

L'évaluation de l'influence des temps de calcul des fonctions d'adaptation pourra se faire à l'aide du graphe un.

De plus pour mieux évaluer l'efficacité des algorithmes testés, les graphes de référence optimisés devront contenir le minimum de croisements.

Ces deux contraintes justifient que les graphes de référence n'aient pas été construit pas un processus aléatoire.

Résumé des caractéristiques des graphes de référence :

	Isthmes	*Points articulation*	*Sommets*	*Arêtes*
Graphe 1	4	2	30	50
Graphe 2	0	0	12	25
Graphe 3	2	3	20	25
Graphe 4	0	0	26	25
Graphe 5	2	1	15	25

Il est à noter que le graphe un a une « complexité » équivalente à deux graphes 5.

3.1.1.1 Graphe un

Ce graphe est très « équilibré », il est constitué de deux composantes fortement connexes connectées par un isthme.

Il contient de plus deux points d'articulation, ainsi qu'une structure en arbre.

Composé de nombreux sommets, il va notamment permettre de prendre en compte la dynamique de l'algorithme testé.

Il s'agit d'un graphe général, incluant toutes les caractéristiques susceptibles d'être rencontrées dans des graphes réels.

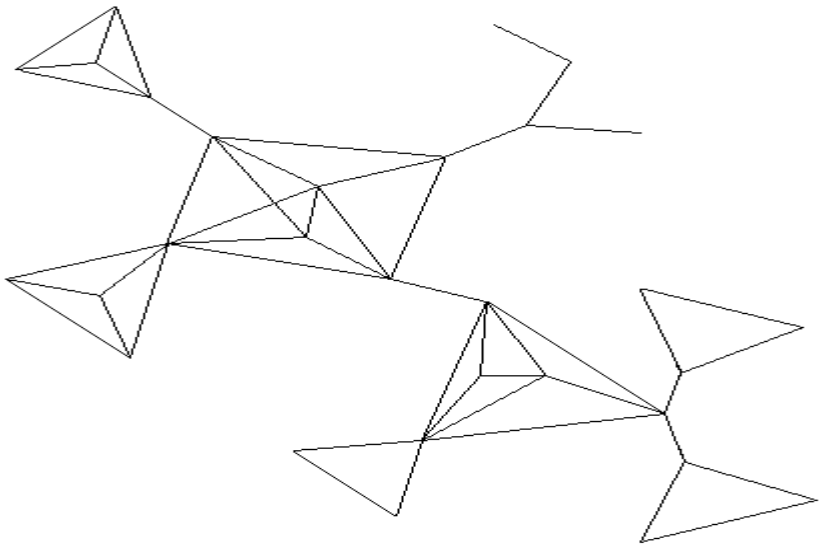


Figure 3-1 : Graphe un, optimisé à 1 croisement.

Ce graphe est constitué de 2 points d'articulation, 4 isthmes, 30 sommets et 50 arêtes.

3.1.1.2 Graphe deux

Ce graphe contient une composante fortement connexe et aucun isthme, point d'articulation ou structure en arbre.

Ce graphe va notamment permettre d'apprécier la capacité d'un algorithme à « démêler » les graphes à forte composante connexe.

Il sera particulièrement utile pour ajuster l'opérateur de reproduction de l'algorithme génétique. L'objectif poursuivi est limiter la scission des schèmes de longueur utile importante.

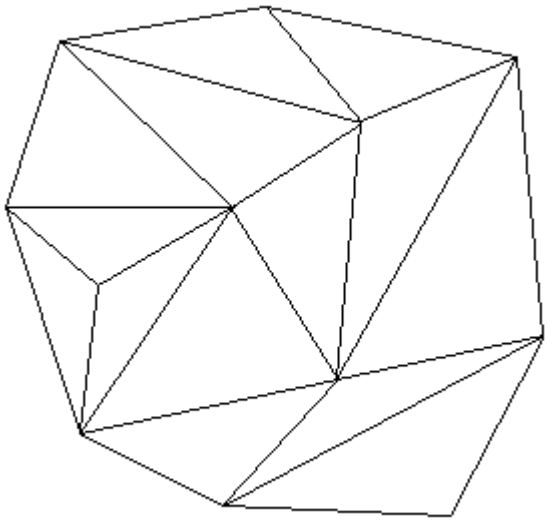


Figure 3-2 : Graphe deux, optimisé à 0 croisement.

Ce graphe est constitué de 12 sommets et 25 arêtes.

3.1.1.3 Graphe trois

Ce graphe est constitué d'isthmes et de points d'articulation connectant des composantes faiblement connexes.

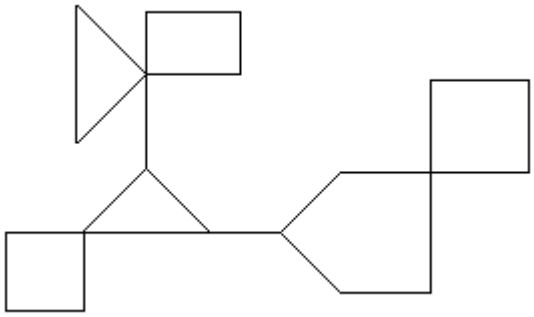


Figure 3-3 : Graphe trois, optimisé à 0 croisement.

Ce graphe est constitué de 3 points d'articulation, 2 isthmes, 20 sommets et 25 arêtes.

3.1.1.4 Graphe quatre

Ce graphe est constitué de structures arborescentes. C'est un cas particulier des graphes susceptibles d'être traités. Il est représentatif d'une structure très hiérarchisée.

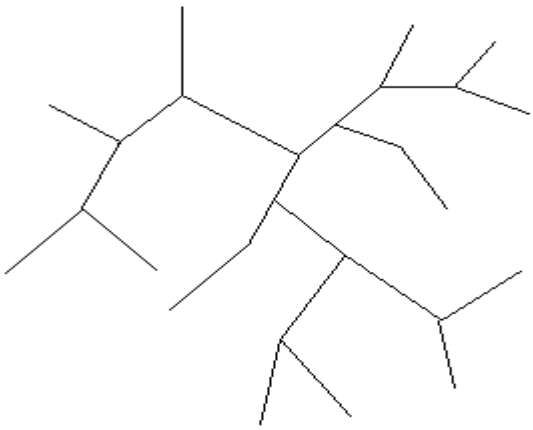


Figure 3-4 : Graphe quatre, optimisé à 0 croisement.

Ce graphe est constitué de 26 sommets et 25 arêtes.

3.1.1.5 Graphe cinq

Ce graphe est lui aussi très équilibré. Il s'agit du graphe exploité comme exemple dans les paragraphes précédents.

Il va permettre d'apprécier la performance globale de la chaîne de traitement. Utilisé conjointement au graphe présenté au paragraphe 3.1.1.1, il va permettre d'apprécier l'influence de la dimension du graphe sur la chaîne de traitement.

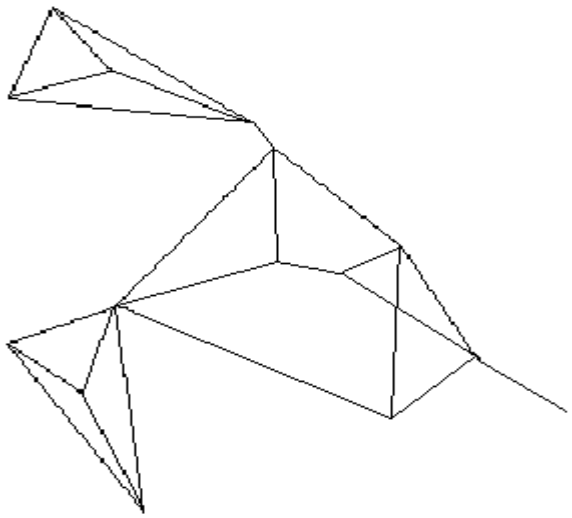


Figure 3-5 : Graphe cinq optimisé à 0 croisement.

Ce graphe est constitué d'un point d'articulation, 2 isthmes, 15 sommets et 25 arêtes.

3.1.1.6 Graphes complémentaires

Les graphes présentés plus haut ont été codés essentiellement en vue de valider la capacité de la chaîne de traitement à optimiser l'ensemble des conformations des graphes susceptibles d'être obtenues dans le monde « réel ».

De nombreux autres graphes ont été exploités pour la mise au point des algorithmes stochastiques, pour affiner les paramètres pilotant l'algorithme de recuit simulé, ainsi que pour la définition des opérateurs retenus dans l'algorithme génétique final.

3.1.2 Structure des données communes à l'ensemble des algorithmes

Comme cela a été précisé plus haut, les bénéfices apportés par la composante déterministe de la chaîne de traitement seront exploités par l'algorithme génétique final.

Il est donc important d'exploiter une codification du graphe à optimiser commune à l'ensemble de la chaîne de traitement. Celle-ci sera simplement constituée par une chaîne construite par la concaténation des coordonnées x et y des différents sommets composant le graphe.

L'ordre d'apparition des sommets dans la chaîne sera déterminé initialement par un classement appliqué lors de l'identification des isthmes et points d'articulation.

Il sera ensuite modifié par l'algorithme génétique appliqué en fin de chaîne de traitement. Le recuit simulé ne modifiant lui que les positions des sommets dans l'espace de tracé, donc les valeurs des couples x et y et non leurs positions dans la chaîne de codage des graphes.

Chaîne de caractères exploitée pour coder le graphe :

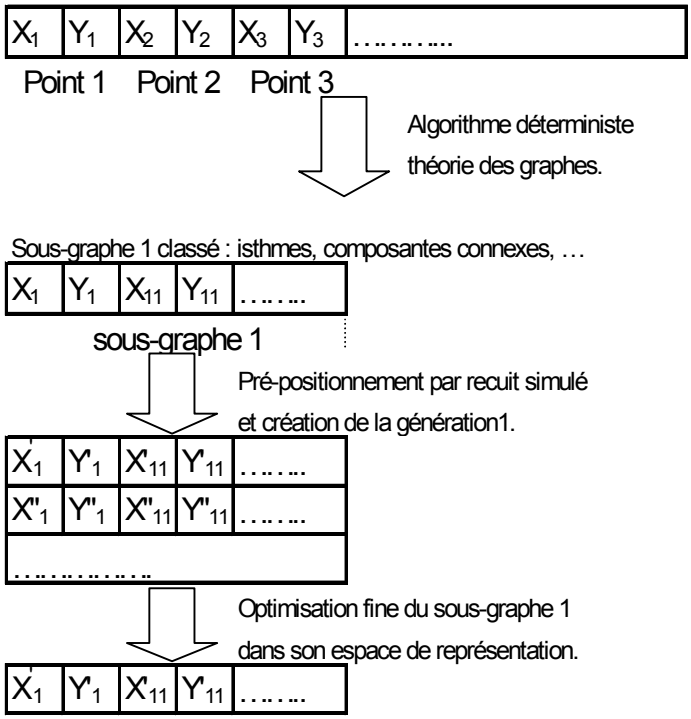


Figure 3-6 : Evolution de la structure de données exploitée.

De plus, l'objectif de l'ensemble des algorithmes appliqués dans la chaîne de traitement est d'optimiser un critère d'esthétisme qui a été limité, dans le cadre de ce travail, au nombre de croisements d'arêtes observé dans une conformation particulière d'un graphe.

Le critère d'esthétisme étant unique, il est important de disposer d'une structure de données globale permettant de l'évaluer.

Pour cela une liste d'adjacence a donc été exploitée, maintenant la définition de l'ensemble des arêtes du graphe.

Cette approche minimise les espaces mémoire de stockage et simplifie le code de la fonction d'évaluation du critère d'esthétisme, ce code étant commun aux algorithmes de recuit simulé et génétiques.

3.1.3 Statistiques et expérimentations :

La chaîne de traitement intègre une composante déterministe ainsi qu'une composante stochastique.

La composante déterministe n'est pas concernée par ce paragraphe, en effet il s'agit fondamentalement d'une méthode fidèle. Appliquer un grand nombre de fois cet algorithme à un même graphe présenté sous des conformations différentes produira systématiquement le même résultat.

La composante stochastique, avec une même conformation de paramètres, pourra au contraire produire des résultats fluctuants.

Il va donc être nécessaire dans cette phase de validation, d'appliquer le même couple paramètres/algorithme à un même graphe, un nombre de fois « suffisant », nous permettant d'apprécier au mieux les performances du couple.

Par contre la grandeur à observer étant produite par lecture directe d'une valeur stockée dans une variable informatique et celle-ci étant calculée par un algorithme déterministe (nombre de croisements de segments de droites), il ne sera pas ici nécessaire d'intégrer les notions courantes aux phases d'expérimentation que sont les erreurs systématiques et aléatoires. La méthode de mesure du résultat final est considérée comme parfaitement fiable et reproductible.

Cinq graphes caractéristiques ont été retenus dans cette phase de validation. Parmi eux le graphe numéro deux qui comporte le moins de sommets. Avec ses douze sommets il pourra donc être représenté dans sa forme la plus confuse par des arrêtes ayant N(N-1)/2 croisements, soit 66 croisements. Alors que ce même graphe est susceptible d'être représenté sans aucun croisement dans sa conformation optimisée. Il existe donc 66 classes de conformations différentes, qui vont autoriser la mise en œuvre d'une approche statistique de l'appréciation de la performance de la composante stochastique de la chaîne de traitement. Du moins quand les conditions d'arrêt n'intègrent pas la notion de seuil de croisement final.

La performance de l'algorithme stochastique va donc correspondre au nombre de croisements obtenus en fin de traitement, nombre de croisements moyenné

sur un nombre d'exécutions suffisant de la chaîne de traitement. La valeur de l'écart-type des nombres de croisements successivement observés au final constituera un indicateur de robustesse du traitement.

La notion de valeur aberrante est volontairement écartée, en effet les conformations particulières, susceptibles d'être produites sont elles aussi représentatives de la fidélité de la chaîne de traitement.

Plusieurs applications d'un même algorithme vont donc produire, pour un même graphe, des conformations de graphes différentes dont le nombre de croisements sera aléatoire, la distribution de ceux-ci pouvant être considérée comme répondant à une distribution normale.

Déterminer un nombre suffisant d'exécution peut être assimilé à déterminer la taille convenable d'un échantillon.

Les formules de calcul suivantes seront appliquées :

Calcul de la moyenne estimée :

$$\bar{x}=\frac{\sum xi}{n}$$

Calcul de l'écart-type :

$$s=\sqrt{\frac{\sum(\bar{x}-x)^2}{n-1}}$$

Limites de confiance d'une nouvelle mesure x_i :

$$\bar{x}-ts<x_i<\bar{x}+ts$$

Le tableau ci-dessous fournit les valeurs de t, permettant d'attribuer un intervalle de confiance à toute nouvelle mesure, avec des probabilités de 95 et 99 % :

N	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***20***	***inf.***
t (95 %)	12,7	4,3	3,2	2,8	2,6	2,5	2,4	2,3	2,3	2,2	2,1	1,96
t (99 %)	63,6	9,9	5,8	4,6	4	3,7	3,5	3,4	3,3	3,2	2,8	2,6

N est le degré de liberté, à savoir le nombre de mesures effectuées moins un.

Le nombre d'exécutions successives d'un même algorithme à effectuer est :

$$n=\frac{t^2 p(1-p)}{h^2}$$

Si l'on considère que le nombre d'exécution d'un même algorithme sera très supérieur à 20, une valeur de t fixée à 1,96 va représenter un risque d'erreur de 5% ce qui est acceptable. Le paramètre p est représentatif du pourcentage d'appartenance d'une valeur à l'intervalle de confiance. Fixer cette valeur à 50% constitue la condition la plus défavorable car p est multiplié par (1-p).

Le nombre d'exécutions d'un même algorithme à effectuer va donc être de :

$$n=\frac{1{,}96^2*0{,}5(1-0{,}5)}{0{,}07^2}=196$$

Donc pour un intervalle de confiance de 7%, environ 200 exécutions du même couple algorithme/paramètres seront à effectuer.

3.2 LES DIFFERENTS ALGORITHMES TESTES

D'autres algorithmes ont été testés avant d'aboutir à cette chaîne de traitement.

3.2.1 Positionnement aléatoire

D'autres algorithmes aléatoires ont été testés dans le cadre de ce travail, avec il faut le dire peu de succès.

L'algorithme qui a donné les meilleurs résultats a consisté à positionner aléatoirement les sommets d'un graphe dans un espace à trois dimensions. Ensuite par le biais d'une caméra mobile (matrices de rotation), des prises de vue ont été faites sous différents angles. La « photographie » (projection sur un plan) offrant le minimum de croisements étant alors retenue.

Cette approche n'a pas fourni de résultats satisfaisants du fait de sa très mauvaise reproductibilité et n'a donc pas été retenue. Néanmoins, elle peut être exploitée pour traiter très rapidement des graphes comportant peu d'arêtes.

3.2.2 Heuristique de Eades

Cet algorithme a été testé.

Il est plus rapide que les algorithmes génétiques ou de recuit simulé exploités plus haut, par contre il ne permet pas d'optimiser des graphes complexes contenant des composantes cycliques connectées par des sous-graphe de type arbre hiérarchique.

3.2.3 Algorithme de Kamada et Kawai

Cet algorithme, a aussi été testé.

Pour cela un code source fourni avec un environnement de développement Java a été exploité.

Le résultat obtenu est très impressionnant car cet algorithme est très dynamique pour des graphes de petite dimension.

Les graphes sont correctement séparés en sous-graphes disjoints, par contre comme l'algorithme précédant, il éprouve quelques difficultés à traiter des graphes complexes, notamment composés de plusieurs composantes cycliques.

De plus il semble important d'implémenter une extension à cet algorithme permettant de mieux maîtriser les dimensions de l'espace de tracé des graphes.

Malgré ces inconvénients, cet algorithme a été implémenté dans la version 2.b du produit MATRISME. Il est susceptible d'être lancé à la demande.

3.2.4 Réseaux de neurones

Dans un précédant travail, j'ai dirigé un projet ayant pour objectif l'exploitation de réseaux de neurones comme modèles de prévision de grandeurs physiques.

Le principe était de mettre en place une méthode déterministe permettant de sélectionner la conformation optimale d'un réseau de neurones, parmi une famille de réseaux de typologie proche (structure et fonction d'activation).

Ce travail a notamment débouché sur les observations suivantes :

- L'absence d'un seul paramètre d'entrée paralyse le traitement neuronal.
- La nécessité d'un corpus d'apprentissage très important.

J'ai tenté d'adapter ce travail à la problématique de l'optimisation des graphes.

Partant de l'hypothèse qu'une approche analytique de la structure des graphes autorisait la mise en évidence de caractéristiques (nombre de sommets, nombre d'arêtes, nombre d'isthmes, nombre de points d'articulation et autres critères de complexité du graphe) permettant d'envisager une classification.

J'ai alors entrepris de rechercher un réseau neuronal permettant d'orienter le pré-traitement des graphes.

Cette approche n'a pas été concluante.

Par contre, si les réseaux neuronaux ne semblent pas adaptés pour résoudre l'optimisation de la représentation des graphes, l'inverse n'est pas vrai.

En effet la chaîne de traitement définie dans le cadre de ce travail est tout à fait capable d'optimiser la représentation des réseaux de neurones. Et notamment les réseaux ayant des conformations proches de la machine de Boltzmann avec des connexions quasi complètes et symétriques.

3.3 LA SOLUTION RETENUE

3.3.1 Découpage du graphe par algorithme déterministes

3.3.1.1 Structure de données exploitée

Les principales structures de données exploitées sont :

- La liste des sommets. Il s'agit d'un tableau de structures constitué des libellés courts et longs des points, des coordonnées X et Y, de l'intervalle de variation des coordonnées des points. Cette liste de sommets va être découpée en sous-listes de sommets appartenant à chacun des sous-graphes. Elle sera ensuite exploitée par l'algorithme de recuit simulé comme base de calcul de la fonction de coût. Et enfin elle constituera la chaîne de caractères codant chacun des sous-graphes optimisés par l'algorithme génétique.
- La liste d'adjacence. Il s'agit d'un tableau de structures constitué des libellés courts et longs des arêtes, des références des deux sommets que l'arête relie ainsi que d'une valeur affectée à l'arête représentative de l'intensité de la liaison.

Ces deux structures ont été réduites à leur plus simple expression. Seules ces deux structures seront exploitées, propageant ainsi les bénéfices apportés par chacun des algorithmes appliqués tout au long de la chaîne de traitement.

3.3.1.2 Algorithmes retenus

3.3.1.2.1 Identification des sous-graphes

L'algorithme retenu pour identifier les sous-graphes est extrêmement simple. D'une part, il offre l'avantage de n'exploiter que la liste d'adjacence initialisée au chargement du graphe, ce qui limite le volume de données à traiter par le programme.

D'autre part, si N est le nombre de sommets du graphe, les temps de traitement peuvent être considérés comme étant d'ordre N, ce qui constitue un traitement exploitable sur un ordinateur personnel.

L'algorithme retenu peut se résumer à :

- Affecter le premier sommet du graphe au premier sous-graphe.

- Exploiter la liste d'adjacence du graphe et identifier les sommets qui seront liés à des sommets appartenant déjà à un sous-graphe identifié.
- Chaque scrutation de la liste d'adjacence ne donnant lieu à aucune affectation d'un point à un des sous-graphes déjà identifié va engendrer la création d'un nouveau sous-graphe et l'affectation d'un premier sommet non affecté à celui-ci.
- Ces traitements sont répétés tant que l'ensemble des sommets constituant le graphe n'a pas été traité.

Les traitements prennent fin avec l'affectation du dernier sommet à un des sous-graphes déjà identifié ou à un nouveau sous-graphe. Il s'agit d'un processus itératif où la liste d'adjacence est scrutée un nombre de fois au maximum égal au nombre de sommets du graphe. La figure ci-dessous illustre l'évolution des temps de calcul de la chaîne de traitement pour les graphes comportant successivement 45 sommets, 30 sommets et 15 sommets.

La figure 3.7 démontre clairement que l'amélioration des temps de calcul apportée par le découpage des graphes en sous-graphe est supérieure à l'ordre N. Il est donc important que la prise en charge de cette problématique se fasse par approche déterministe.

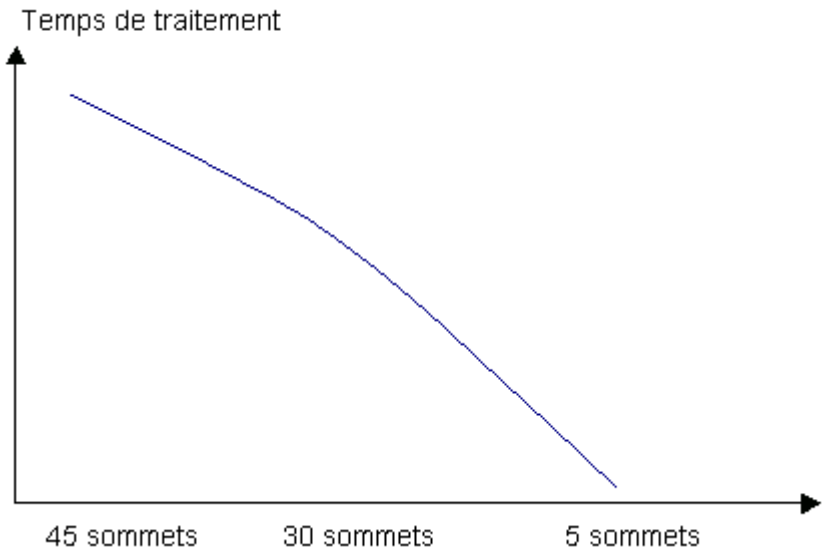


Figure 3-7 : Découpage des graphes en sous-graphe.

3.3.1.2.2 Principe d'affectation d'espaces de représentation distincts

L'espace d'affichage va être scindé en sous-espaces par ajouts successifs alternativement d'une séparation verticale, puis d'une séparation horizontale jusqu'à l'obtention d'un nombre suffisant d'aires de représentation, comme l'illustre la figure suivante :

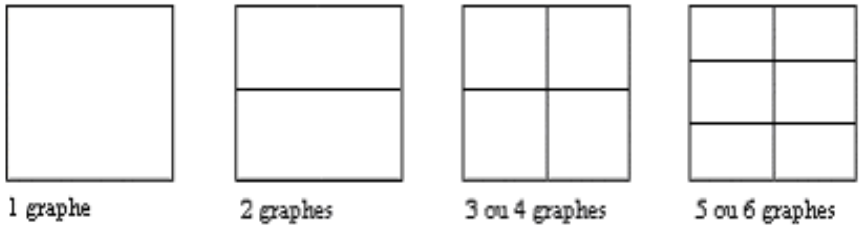


Figure 3-8 : Découpage de l'espace de tracé des graphes.

Sur la copie d'écran de l'application développée dans le cadre de ce travail de recherche, fournie ci-dessous, nous pouvons constater la séparation de l'espace de tracé, les deux graphes étant correctement affectés à des zones distincte. Cette première organisation de positionnement d'un graphe contribue déjà à une amélioration de la compréhension de l'information qu'il recèle :

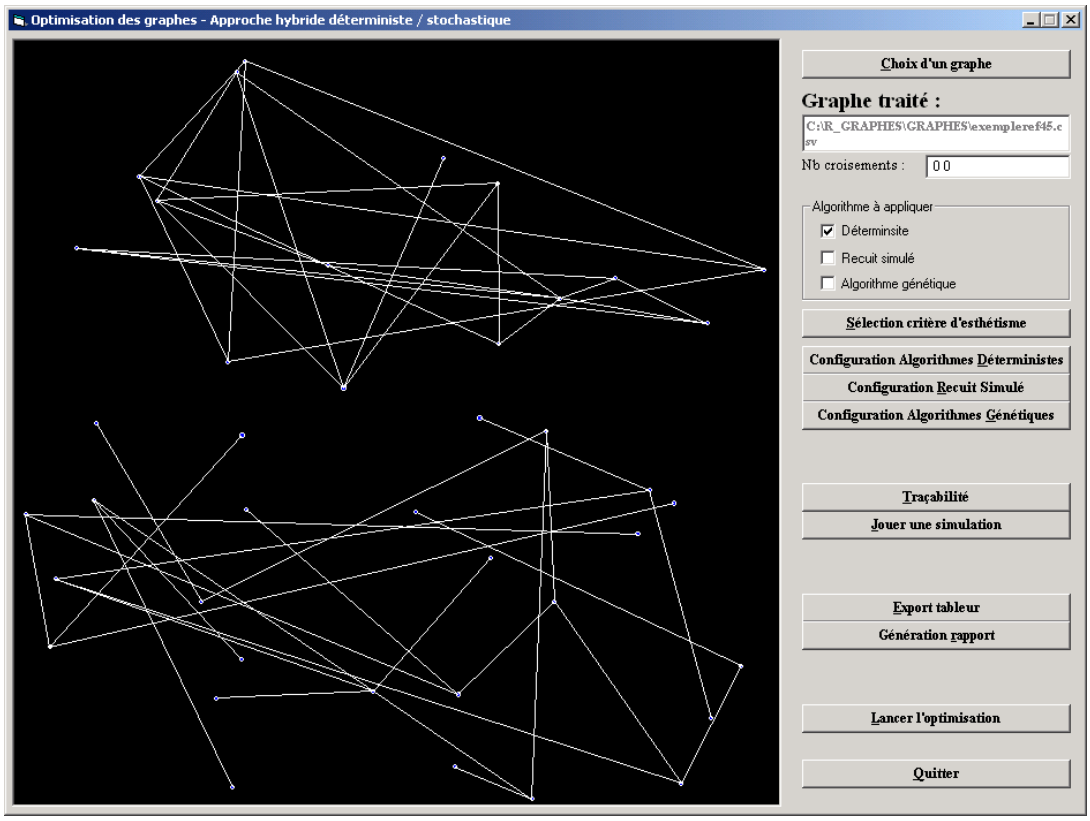


Figure 3-9 : Graphes 4 et 5 correctement dissociés.

L'algorithme retenu est le suivant :

```
TantQue NombreDeZoneDeTracé < NombreDeSousGraphe
    SI Pas(PassagePair) Alors
        PassagePair = vrai
        NombreDeZoneDeTracé = NombreDeZoneDeTracé * 2
        NombreHorizontale = NombreHorizontale + 1
    Sinon
        PassagePair = faux
        NombreDeZoneDeTracé = NombreDeZoneDeTracé * 2
        NombreVerticale = NombreVerticale + 1
    FinSi
FinTantQue
```

Cet algorithme n'est pas très élaboré, il ne prend pas en compte le nombre de sommets contenus dans chacun des sous-graphes. Il est cependant suffisant pour valider la capacité de la chaîne de traitement à identifier les sous-graphes. Les zones de tracé obtenues sont de forme carrée, ceci permet d'envisager un positionnement immédiat des sommets sans avoir à appliquer d'autres traitements qu'une mise à l'échelle et deux translations.

3.3.1.2.3 Identification des isthmes et des points d'articulation

L'algorithme détaillé plus haut, permettant le découpage du graphe en sous-graphes va être à nouveau mis à contribution.

L'identification des isthmes se fait par suppression successive de chacune des arêtes des sous-graphes mises en évidence. Si une arête supprimée engendre un sous-graphe supplémentaire, alors celle-ci peut être considérée comme étant un isthme.

L'identification des points d'articulation va s'appuyer sur le même type de raisonnement, si la suppression d'un des sommets du graphe engendre un sous-graphe supplémentaire, alors le point supprimé pourra être considéré comme étant un point d'articulation.

Si N est le nombre de sommets du sous-graphe analysé, les temps de traitement peuvent être considérés comme étant d'ordre N^2 pour chacun de ces deux algorithmes, le surcoût en temps de calcul représenté par cette optimisation reste donc acceptable.

Le graphe de référence numéro trois, qui contient des isthmes et des points d'articulation a été exploité comme base d'expérimentation.

Les améliorations éventuellement apportées à l'algorithme de recuit simulé n'ont pas été quantifiées, en effet ce pré-traitement a pour objectif de commencer à organiser la structure informatique abritant le graphe à optimiser et non à pré-positionner les différents sommets dans un espace de tracé défini.

La figure ci-dessous illustre les améliorations apportées à l'algorithme génétique par le découpage en isthmes et points d'articulations.

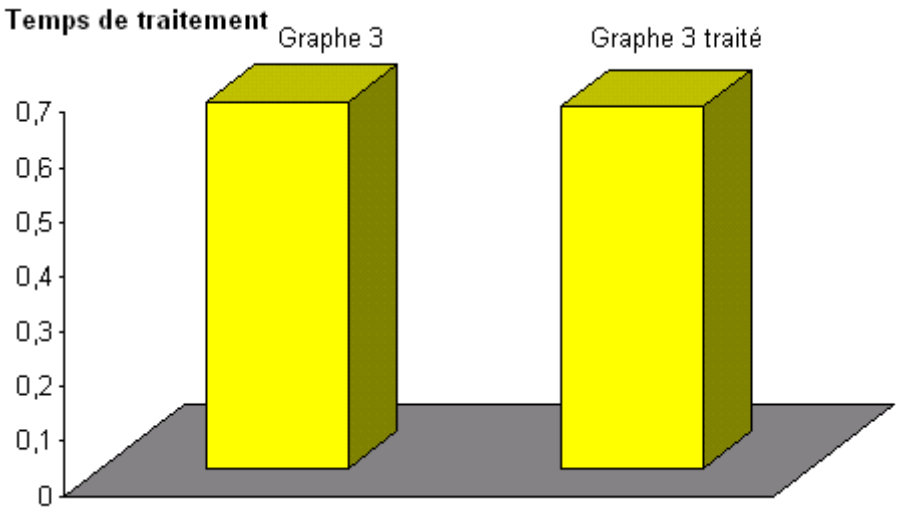


Figure 3-10 : Bénéfices de l'identification des isthmes et des points d'articulation.

Ce graphique a été obtenu en moyennant les temps de traitement observés par 200 applications successives d'un algorithme génétique simplifié. La durée maximale observée durant ces essais a été ramenée à l'unité (ce qui permet de s'affranchir de la puissance de calcul de l'ordinateur exploité).

Seuls les opérateurs de reproduction et de croisement ont été retenus lors de cet essai.

La transformation linéaire de la fonction d'adaptation a été exploitée, celle-ci étant bornée à deux fois la moyenne des fonctions d'adaptation.

L'opérateur de reproduction appliqué a mis en œuvre une sélection par roue de loterie sans autre correction.

3.3.1.2.4 Classement des sommets

Les algorithmes permettant d'identifier les points d'articulation ainsi que les isthmes ont été mis en œuvre uniquement pour classer les sommets en vu d'optimiser l'algorithme génétique appliqué ultérieurement.

Les bénéfices apportés par ce classement sont illustrés par la figure 3.10. L'algorithme retenu ne sera pas détaillé ici, en effet celui-ci a peu d'intérêt. Il consiste simplement à regrouper les sommets constituant les composantes connexes, au sein des différentes chaînes de caractères codant les sous-graphes.

Cette démarche initie un début d'organisation des chaînes de caractères codant les sous-graphes, elle est susceptible de mettre en évidence certains schèmes qui pourront ensuite être maintenus et développés tout au long du traitement par algorithmes génétiques.

Une optimisation supplémentaire a été mise en œuvre ici, elle consiste à éloigner les composantes connexes de faible dimension (schèmes courts) dans la chaîne de caractères. En effet ceux-ci sont théoriquement sensés être mieux pris en compte par les algorithmes génétiques.

Cette optimisation a été maintenue dans la chaîne de traitement finale car elle est peu coûteuse en temps de traitement et ne peut qu'engendrer une amélioration de la performance des algorithmes génétiques.

Par contre l'expérimentation sur les graphes de référence n'a pas permis de démontrer une amélioration flagrante de l'efficacité des traitements tant en terme de rapidité que de capacité à organiser les graphes complexes (graphe de référence n°1).

3.3.1.3 Résultats observés

Comme l'illustre la figure 3.9, représentant le tracé du graphe regroupant les graphes de référence 4 et 5, les différents sous-graphes sont correctement séparés.

Des espaces de tracé distincts leur sont affectés. L'intérêt de scinder le graphe en sous-graphe est évident.

Le pré-positionnement, des composantes connexes liées par des isthmes ou des points d'articulation, dans l'espace de tracé n'a pas été maintenu. Comme le montre la figure 3.11, chaque sous-graphe doit être optimisé globalement par les algorithmes stochastiques car l'optimisation d'un critère d'esthétisme peut impliquer le déplacement complet d'une composante fortement connexe.

Cette possibilité de réorganisation complète du sous-graphe doit être laissée aux algorithmes stochastiques.

En d'autres termes les algorithmes stochastiques ne seront pas appliqués sous contrainte, certains points ayant des positions prédéfinies dans l'espace de tracé.

Par contre l'identification des points d'articulation et des isthmes a été maintenue. Ce découpage qui tend à organiser le codage du sous-graphe en cours de traitement a un intérêt en relation avec l'algorithme génétique.

Ces opérateurs permettent d'initialiser de manière intelligente la première génération d'individus traitée par algorithme génétique et ce en relation avec la notion de schèmes développée au paragraphe 2.6.8.

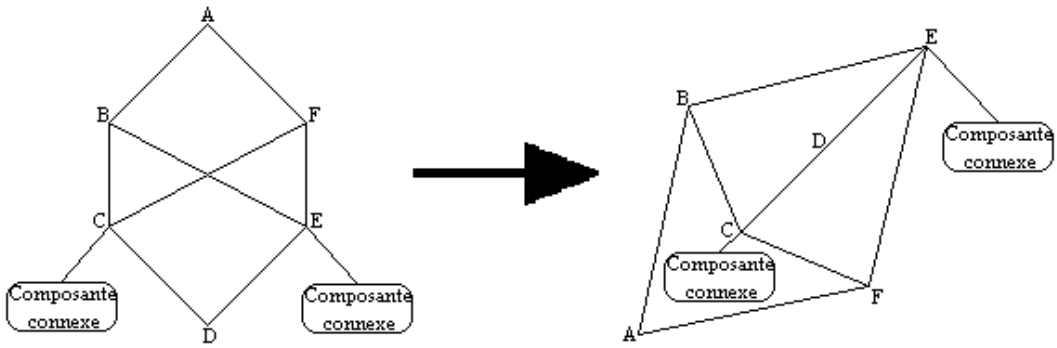


Figure 3-11 : Déplacement des composantes connexes.

3.3.1.4 Durée des traitements

Comme nous l'avons vu aux paragraphes précédents les temps de calcul des algorithmes déterministes sont quantifiables, de l'ordre N ou N^2, il s'agit donc de traitements légers tout à fait adaptés à l'environnement informatique envisagé dans le cadre de ce travail.

De plus, la première génération traitée par les algorithmes génétiques aura déjà une certaine organisation, ce qui va améliorer la convergence globale de la chaîne de traitement.

Une partie de la charge de calcul liée à notre problématique est ainsi transférée de l'algorithme génétique qui est un algorithme lourd et coûteux en temps de calcul vers un algorithme déterministe plus léger.

3.3.1.5 Algorithme déterministe définitif retenu

- Initialisation des deux structures de données :
  - ✓ Liste des sommets (chaîne de caractères).
  - ✓ Liste d'adjacence.
- Découpage en sous-graphes, détermination d'espaces de tracé distincts.
- Identification des isthmes et points d'articulation.
- Classement des sommets, regroupement des sommets appartenant à des composantes connexes.

3.3.2 Appréciation du critère d'esthétisme

Comme nous l'avons vu plus haut, les algorithmes probabilistes exploités ont pour objectif d'optimiser une fonction caractéristique de l'esthétisme d'un graphe.

Cette fonction pourrait être la combinaison linéaire d'un ensemble complexe de critères différents, pourtant dans le cadre de ce travail seule la notion de nombre de croisements d'arêtes a été retenue.

Ce critère semble le plus naturel, il met en évidence la structure des informations contenues dans le graphe et est synonyme d'une amélioration de la lisibilité.

L'algorithme permettant de calculer le nombre de croisements d'arêtes contenus dans un graphe est détaillé ci-dessous.

Il a pour principal intérêt de s'appuyer sur les deux structures de données, communes à l'ensemble des algorithmes d'optimisation, d'où l'économie d'étapes supplémentaires de codage des graphes implémentés à chaque changement de méthode d'optimisation.

De plus il prend en compte les intersections entre arêtes verticales et horizontales, ce qui n'est pas le cas d'autres algorithmes fournis par la littérature, plus rapides mais moins précis.

> *Calcul des équations des deux droites superposées aux arêtes.*
>
> *Si le calcul du point d'intersection des deux droites et possible (droites non parallèles) et si le point d'intersection se situe sur une des deux arêtes, alors il y a croisement.*

Le code de cet algorithme a été localisé dans une fonction totalement détachée. Le paramètre d'entrée de la fonction est constitué du graphe à évaluer. La fonction retourne le nombre de croisements calculé.

Cette organisation du code source permettra donc de faire évoluer le critère d'esthétisme de manière complètement indépendante aux différents algorithmes d'optimisation retenus.

3.3.3 Pré-traitement des graphes par recuit simulé

3.3.3.1 Structure de données exploitée

La mise en œuvre de cet algorithme ne nécessite pas de définition d'une structure de données supplémentaire.

Les listes d'adjacence définies plus haut ainsi que les listes de sommets sont directement exploitées par l'algorithme de recuit simulé, ce qui en soit constitue déjà une certaine optimisation.

3.3.3.2 Algorithme définitif retenu

L'attribution de zones de tracé distinctes pour chacune des composantes connexes connectés par des isthmes ou des points d'articulation a été envisagée. La position des isthmes et points d'articulation étant figée il était alors possible d'optimiser individuellement les parties fortement connexes du graphe par le biais d'algorithmes probabilistes appliqués sous contrainte.

Comme le précise le paragraphe 3.1.5.4, cette démarche n'a pas abouti.

En effet il est préférable de tracer les sous-graphes dans des espaces de tracé distincts et ensuite d'optimiser globalement chaque sous-graphe par la chaîne de traitement composée d'un algorithme de recuit simulé complétée par un algorithme génétique.

La fonction de coût exploitée est la traduction directe du critère d'esthétisme retenu, dans la mesure ou notre objectif est de la minimiser. Là encore aucun calcul supplémentaire visant à coder le critère d'esthétisme n'est à envisager.

3.3.3.3 Paramètres généraux et conditions d'arrêt

3.3.3.3.1 Paramètres retenus

De nombreuses variantes d'algorithmes de recuit simulé sont détaillées dans la littérature.

La mise au point de cet algorithme va consister à déterminer la variante la mieux adaptée à notre problématique ainsi que son paramétrage.

3.3.3.3.1.1 Température initiale

La littérature ne fournit pas de méthodes permettant de définir la température initiale.

Celle-ci est le plus souvent déterminée de manière empirique. Néanmoins Heckenroth [Heckenroth-90] a exploité l'équation suivante :

- $T0 = \ln(r0)/- \Delta E$, avec $r0 = 0,5$ et ΔE fixé en fonction de la complexité du graphe.

ΔE (la variation d'énergie du système) est alors fixé empiriquement, il peut être assimilé au nombre moyen de croisements apparaissant ou disparaissant après avoir déplacé un des sommets du graphe. Il est systématiquement recalculé pour chacun des graphes à optimiser.

Le principe retenu est de générer aléatoirement différentes conformations du graphe et d'enregistrer l'évolution du critère d'esthétisme suite à l'application d'évolutions mineures de conformation.

Les tests effectués sur les graphes de référence ainsi que sur des graphes plus complexes ont montré que déplacer aléatoirement cinq fois le sommet d'un graphe était suffisant pour apprécier grossièrement ΔE.

La figure ci-dessous exprime l'évolution de ΔE pour le graphe de référence numéro un :

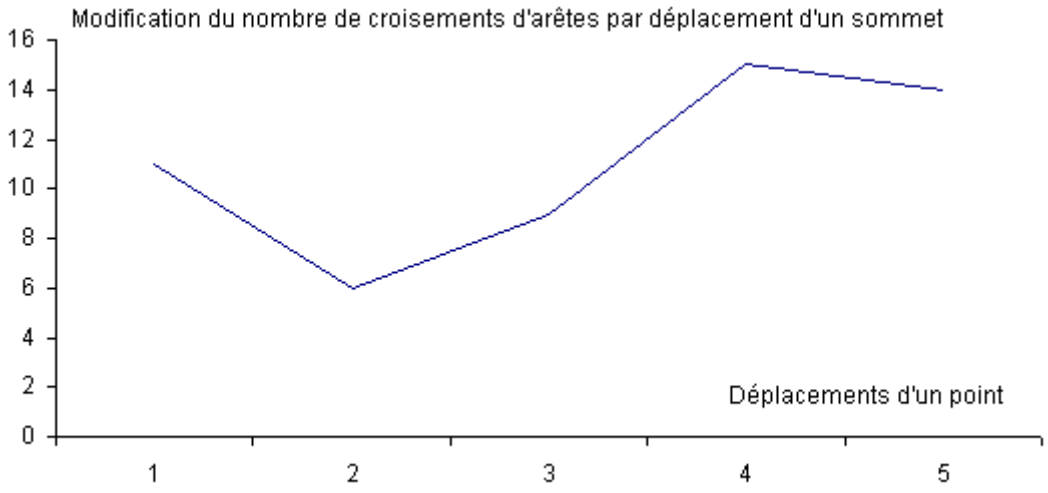


Figure 3-12 : Evolution de ΔE.

On observe ici, mais cela s'est révélé vrai pour l'ensemble des graphes de référence, que modifier aléatoirement la position d'un des sommets d'un graphe engendre une variation maîtrisée du critère d'esthétisme.

De plus en retenant comme fonction d'évolution de la température une décroissance géométrique de raison 0.9, il a été constaté qu'appliquer un facteur multiplicatif à la température initiale n'engendrait qu'une faible augmentation des temps de traitement. L'application d'un facteur multiplicatif de la température initiale a donc été retenue car il permet d'améliorer l'exploration de l'espace de recherche.

Ce paramètre est lié au graphe à optimiser, par contre il est indépendant de la conformation du graphe à optimiser.

3.3.3.3.1.2 Evolution de la température

La température évolue généralement par paliers. Plusieurs approches sont susceptibles d'être retenues [Alaoui-93] :

- Décroissance géométrique de raison s, classiquement fixée à 0.9.
- $T_n = T_0 \exp(-k(n-1))$, avec k constante telle que pour $T_0 = 100$, la température finale atteigne $T_{50} = 10^{-6}$ pour n = 50 itérations.

Ces deux approches ont été testées. Le graphe ci-dessous exprime le parcours des paliers de température pour chacune des approches.

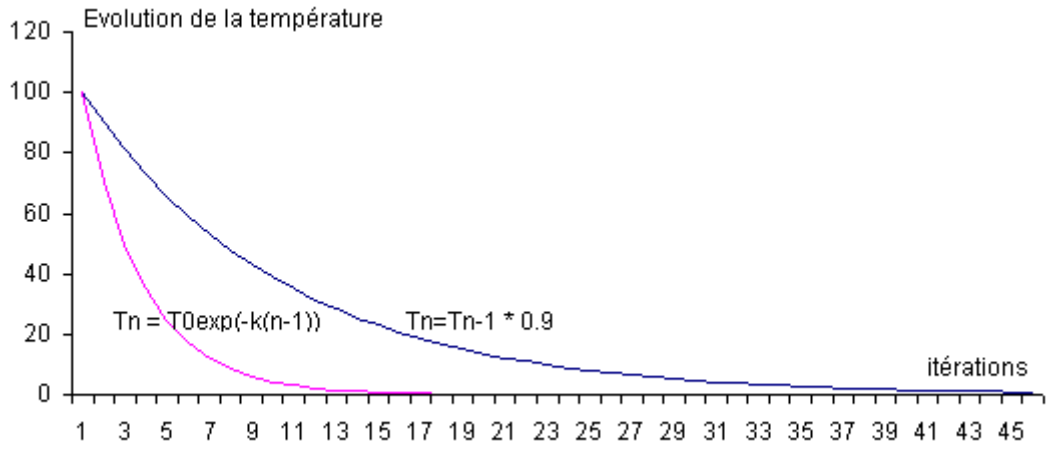


Figure 3-13 : Evolution de la température.

L'application d'une décroissance géométrique a été finalement retenue. Il apparaît clairement que la diminution de la température est plus progressive.

De plus des tests ont montré qu'en appliquant comme modification élémentaire un déplacement maîtrisé (le vingtième de l'intervalle de l'espace de tracé) d'un des sommets du graphe, l'application d'une décroissance géométrique plutôt

qu'une diminution exponentielle de la température, « laissait » plus de temps au graphe pour s'organiser.

L'application d'une diminution exponentielle de la température engendre très rapidement des paliers de températures inefficaces ou l'ensemble des nouvelles modifications de conformations sont rejetées. Ceci correspond en fait à des temps de calcul qui ne sont pas susceptibles d'apporter d'améliorations tant en terme de parcours de l'espace de recherche que de capacité qu'ont les algorithmes de recuit simulé à sortir d'un minima local.

Retenir la décroissance géométrique comme fonction d'évolution des paliers de température constitue une double optimisation des temps de traitement. Non seulement on observe moins de calculs inefficaces, mais de plus cette fonction est plus simple donc moins consommatrice de temps de calcul.

3.3.3.3.1.3 Modifications élémentaires du graphe

Ce sont ces modifications qui sont à la base de l'évolution de la conformation d'un graphe. L'objectif est d'apporter à température constante, une légère modification du système, impliquant une modification du critère d'esthétisme qui sera conservée si elle améliore celui-ci, ou qui sera rejetée s'il y a dégradation du critère d'esthétisme et ce avec une probabilité de plus en plus important au fur et à mesure que la température diminue.

Les modifications élémentaires susceptibles d'être appliquées à un graphe sont les suivantes :

- Modification maîtrisée d'une coordonnée x ou y d'un des sommets du graphe ;
- Modification maîtrisée des coordonnées x et y d'un des sommets du graphe ;
- Modification maîtrisée de l'ensemble des coordonnées x ou y des sommets du graphe ;
- Modification maîtrisée de l'ensemble des coordonnées y et x des sommets du graphe.

Ces modifications doivent être « élémentaires ». En effet l'algorithme de recuit simulé va accepter ou non une modification du graphe parmi en ensemble de modifications mineures dont les effets vont conduire à une conformation optimisée relativement au critère d'esthétisme retenu.

Ces modifications doivent être maîtrisées car on poursuit deux objectifs, le premier est de tendre progressivement vers un état stable du système étudié, le deuxième est de garder la capacité d'appliquer des modifications suffisamment importantes au système pour lui permettre de sortir d'un minima local et ce essentiellement pour des températures élevées.

Les tests effectués ont montré que modifier une seule coordonnée d'un point (x ou y) plutôt que les deux coordonnées x et y simultanément aboutit à un résultat identique. La modification d'une seule coordonnée correspondant à un temps de traitement inférieur, c'est cette démarche qui a été retenue.

Par contre la convergence générale de l'algorithme de recuit simulé s'est avérée meilleure en appliquant des modifications maîtrisées simultanément à l'ensemble des sommets d'un graphe plutôt qu'à un seul de ses sommets.

Ces observations ont été confirmées pour l'ensemble des graphes de référence exploité.

Par contre la détermination de l'intensité de la modification appliquée s'est avérée plus difficile. Les nombreux tests effectués sur les graphes de référence, ainsi que sur des graphes correspondant à des concaténations de graphes de référence m'ont conduit à figer cette valeur. En effet envisager de faire évoluer l'intensité du déplacement des sommets d'un graphe en fonction de l'évolution de la température est une approche séduisante, néanmoins l'expérimentation n'a pas permis d'établir de règles en la matière.

Figer la modification élémentaire des sommets du graphe durant toute l'application de l'algorithme de recuit simulé a été la solution retenue, le déplacement appliqué aux sommets d'un graphe étant proportionnelle à l'espace de tracé.

3.3.3.3.1.4 Test d'équilibre thermodynamique

Heckenroth [Heckenroth-90] fournit un exemple des valeurs des paramètres à mettre en œuvre. Notamment pour atteindre l'équilibre thermodynamique à une température donnée :

- Equilibre thermodynamique atteint dès que 100*n tentatives ont été effectuées ou dès que 12*n modifications ont été acceptées, n étant le nombre de sommets constituant le sous-graphe en cours de traitement.

Les deux tests d'équilibre thermodynamiques ont été évalués.

Toujours dans un souci d'optimisation le test retenu s'est appuyé sur l'application de la règle des 100*n tentatives. L'expérimentation de cette condition d'arrêt sur l'ensemble des graphes de référence à mis en évidence une économie du nombre d'itérations à température constante, notamment pour des températures faibles.

Cette approche va peut-être à l'encontre de la philosophie de l'algorithme de recuit simulé en ne lui laissant pas l'opportunité de continuer d'améliorer la conformation du graphe pour des températures faibles.

Néanmoins il faut considérer que le rôle de cet algorithme est d'initier une certaine organisation du graphe qui sera ensuite affinée par un algorithme génétique.

3.3.3.3.1.5 Température finale

Les conditions couramment exploitées pour arrêter un algorithme de recuit simulé sont soit l'absence de détérioration de la fonction de coût durant un nombre prédéfini de paliers de température, soit la cible « température finale » atteinte.

Le nombre de paliers de températures à explorer est un paramètre redondant avec le paramètre « température finale » car il est lui aussi lié à la température initiale et à la fonction d'évolution de la température.

Les deux conditions d'arrêt citées plus haut ne sont pas incompatibles, c'est pourquoi elles ont été toutes les deux implémentées.

L'expérimentation sur les graphes de référence, ainsi que sur des graphes correspondant à des concaténations de graphes de référence a mis en évidence la nécessité de prendre en compte l'absence de détérioration de la fonction de coût durant un nombre prédéfini de paliers de température.

Cette condition a été essentiellement exploitée pour sortir de l'algorithme de recuit simulé durant la phase d'expérimentation, la cible « température finale » étant rarement atteinte avec la conformation des paramètres retenus.

D'ailleurs implémenter cette condition d'arrêt autorise le surdimensionnement du paramètre température finale, offrant ainsi la possibilité à l'algorithme de recuit simulé de se poursuivre même à des températures très basses, tant qu'il apporte une amélioration du critère d'esthétisme retenu.

Dans le cadre de cette problématique, un autre critère d'arrêt de l'algorithme de recuit simulé a été pris en compte : l'absence de croisement d'arêtes du graphe en cours d'optimisation.

Ce critère d'arrêt a été implémenté après chaque évaluation du critère d'esthétisme, il autorise non seulement l'arrêt de l'algorithme de recuit simulé, mais aussi de l'ensemble de la chaîne de traitement. Il a été implémenté dans le code d'appréciation du critère d'esthétisme.

3.3.3.3.1.6 Condition d'application de l'algorithme génétique

La notion de complexité des graphes traités ne permet pas de définir un critère absolu autorisant ou non l'application de l'algorithme génétique en complément de l'algorithme de recuit simulé mis à part l'absence de croisement d'arêtes observée dans le graphe au sortir de l'algorithme de recuit simulé.

L'algorithme génétique sera donc systématiquement appliqué si le critère d'esthétisme du graphe en cours d'optimisation peut encore être optimisé, celui-ci intègre une condition d'arrêt liée à l'absence d'optimisation du graphe en cours de traitement.

Nous avons vu précédemment que les algorithmes déterministes découlant de la théorie des graphes permettaient d'identifier la présence de graphes planaires par recherche de composantes connexes caractéristiques, mises en évidence par la réduction du graphe en cours de traitement.

Ils permettent simplement d'affirmer qu'un graphe ne comportant que des composantes planaires est susceptible d'être représenté sans croisement d'arêtes et donc que les traitements peuvent se poursuivre tant que cet objectif n'a pas été atteint.

Ces traitements n'ont pas été retenus, en effet d'une part ils ne permettent pas de quantifier l'état d'avancement de l'optimisation en cours et d'autre par ils spécifient la chaîne de traitement, limitant ainsi la mise en œuvre de critères d'esthétisme plus complexes.

3.3.3.3.1.7 Remarque

L'algorithme de recuit simulé est un heuristique, il ne va pas produire le graphe optimal absolu, par contre la dernière conformation de graphe construite pourra être exploitée pour créer la première génération, base d'application de l'algorithme génétique.

Ceci par opposition à l'algorithme génétique qui traite en parallèle un ensemble de conformations d'un même graphe parmi laquelle il faudra extraire la conformation optimale (éventuellement observée sur les n dernières générations).

3.3.3.3.1.8 Expérimentation

La démarche a donc été de s'orienter vers un algorithme de recuit simulé ayant la capacité de s'adapter à la complexité du graphe en cours de traitement. En effet seul le test d'équilibre thermodynamique intègre un nombre de tests finis, pouvant éventuellement ne pas être tous exécutés.

Par contre la température finale susceptible d'être atteinte a été volontairement fixée à une valeur très basse pour permettre ainsi à l'algorithme de poursuivre son action tant que celle-ci est bénéfique.

3.3.3.4 Résultats observés

L'expérimentation a consisté à appliquer uniquement l'algorithme de recuit simulé à chacun des graphes de référence et ce 200 fois successivement. Le nombre de croisements observés pour chacun des runs a ensuite été moyenné.

Il a ainsi été mis en évidence que l'algorithme de recuit simulé apporte une amélioration de l'organisation du graphe traité, sans pour cela atteindre l'optimisation idéale.

3.3.3.4.1 Validation du résultat

Pour valider cela, le nombre moyen d'évaluations du critère d'esthétisme effectuées durant l'application de l'algorithme recuit simulé appliqué à optimiser le graphe de référence numéro un, pendant 200 essais successifs à été calculé, il est noté N.

Ensuite un algorithme aléatoire a été développé, son principe a été de positionner aléatoirement les sommets du graphe de référence numéro un dans l'espace de tracé puis de modifier N fois les positions des sommets selon le principe présenté au paragraphe 3.4.3.1.3. et de ne retenir que les modifications améliorant le critère d'esthétisme.

Cet algorithme aléatoire a été appliqué 200 fois successivement, le nombre de croisements observés pour chacun des runs a ensuite été moyenné puis comparé au résultat obtenu par application de l'algorithme de recuit simulé.

Les bénéfices apportés par l'algorithme de recuit simulé ont ainsi été mis en évidence et ce pour des temps de calcul très inférieurs à ceux couramment observés pour les algorithmes génétiques.

3.3.4 Application de l'algorithme génétique

3.3.4.1 Structure de données

La chaîne de caractères traitée par le recuit simulé pour coder les différentes coordonnées du graphe a été ici exploitée.

Elle a été transformée en tableau de chaînes de caractères pouvant ainsi abriter une génération de graphes.

Quand cela a été nécessaire des structures complémentaires ont été maintenues pour permettre la conservation, sur plusieurs générations, d'individus particulièrement performants.

3.3.4.2 Algorithme définitif retenu

3.3.4.2.1 Création de la génération initiale

Les tests effectués durant la phase de mise au point de l'algorithme de recuit simulé ont démontré que celui-ci améliorait l'organisation des graphes traités. Cette amélioration étant supérieure à celle obtenue par un processus purement aléatoire.

C'est pourquoi cet algorithme sera systématiquement appliqué. Il va produire un graphe déjà optimisé dont les positions des sommets vont être légèrement modifiées pour produire la première génération de graphes prise en charge par l'algorithme génétique.

Plusieurs méthodes de création de la première génération d'individus ont été envisagées :

- Modification maîtrisée d'une des coordonnées x ou y d'un des sommets du graphe ;
- Modification maîtrisée d'une des coordonnées x et y de l'ensemble des sommets du graphe ;
- Modification importante (sur la totalité de l'espace de tracé) d'une des coordonnées x ou y d'un des sommets du graphe ;

Les trois méthodes ont été expérimentées.

Appliquer une modification importante d'une des coordonnées x ou y a un des sommets engendre une perturbation trop importante du graphe qui dégrade l'organisation produite par l'application du recuit simulé. L'algorithme génétique

appliqué ensuite ne recrée par la conformation du graphe obtenue en sortie de recuit simulé, une solution optimale produite en sortie de recuit simulé serait ainsi perdue.

La méthode finalement retenue pour créer la première génération d'individus a été d'appliquer une modification mineure d'une des coordonnées x ou y d'un des sommets du graphe sélectionné aléatoirement.

Pour éviter de trop dégrader le graphe initial, obtenu en sortie de recuit simulé, la modification de coordonnées a été fixée empiriquement à un pourcent de l'espace de tracé total.

Il a été considéré que l'algorithme génétique appliqué par la suite aura une influence majoritaire sur le graphe finalement retenu d'autant qu'il dispose intrinsèquement de la capacité à explorer la totalité de l'espace des solutions possibles.

3.3.4.2.2 Opérateur de croisement sous contrainte

Une tentative de maîtrise de la scission des chaînes de caractères exploitées pour coder les différents individus a été tentée.

L'objectif était de contraindre l'algorithme génétique à croiser de manière intelligente les graphes entre eux en évaluant systématiquement la fonction d'adaptation des nouveaux graphes créés après application d'un croisement positionné successivement sur chacun des caractères constituant l'individu.

Cette technique qui n'a permis d'améliorer que très légèrement la convergence de l'algorithme, n'a pas été maintenue car elle ne compense pas l'augmentation prohibitive des temps de traitement qu'elle engendre.

En effet, l'évaluation de la fonction d'adaptation voit sa fréquence multipliée par le nombre de sommets du graphe, de plus pour être productive cette modification implique que les individus appareillés aient une certaine affinité.

L'application d'un opérateur de croisement uniforme a aussi été prise en compte dans le cadre de cette approche, cette démarche très lourde en temps de calcul n'a pas non plus été maintenue, que les points pivots de l'opérateur de croisement soient choisis relativement aux sous-ensembles connexes déjà identifiés ou qu'ils soient produit par un processus aléatoire.

3.3.4.2.3 Principe de sélection des parents (roulette)

L'algorithme de sélection des parents retenu est du type « Roue de loterie biaisée ». Il n'y a pas lieu de complexifier cet opérateur dans notre domaine d'application.

En effet, d'une part l'optimisation du graphe est appliquée sans contrainte particulière, seule l'appréciation de la fonction d'adaptation oriente le développement ou la récession de certains individus. D'autre part la dimension des générations finalement retenues (deux cents individus) est suffisante pour que la sélection soit représentative de la probabilité de survie de chacun des individus.

Il est à noter néanmoins que plusieurs tours de roue sont effectués avant d'arrêter la sélection.

La figure ci-dessous illustre cet opérateur.

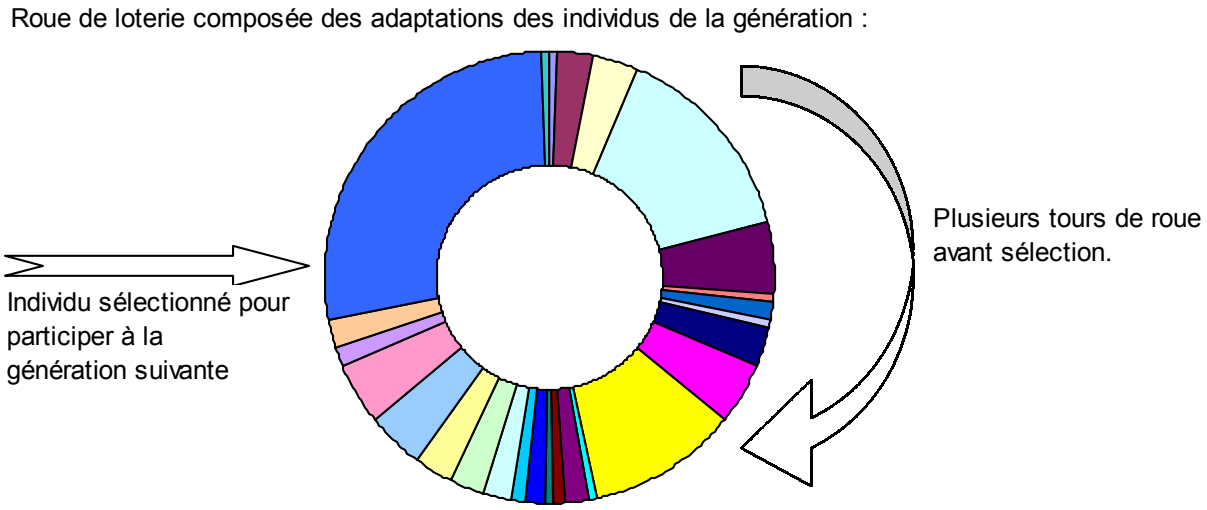


Figure 3-14 : Opérateur de reproduction.

3.3.4.2.4 Calcul de la fonction d'adaptation

Optimiser une fonction revient à trouver sa valeur maximale or optimiser le critère d'esthétisme retenu (nombre de croisements) consiste à rechercher sa valeur minimale. C'est pourquoi la fonction d'adaptation calculée pour chacun des individu va intégrer une légère modification du critère d'esthétisme, sa valeur optimale devant correspondre à sa valeur maximale.

L'optimisation des temps de traitement nous a conduit à intégrer le calcul de la fonction d'adaptation au code prenant en charge la modulation de l'adaptation.

3.3.4.2.5 Modulation de l'adaptation

Comme nous l'avons vu plus haut, n'exploiter que les individus les plus performants pour créer la génération suivante permet d'augmenter la convergence de l'algorithme. Par contre cette démarche peut engendrer la perte prématurée d'une caractéristique intéressante d'un individu globalement non performant, le développement de caractéristiques sans intérêt d'individus globalement intéressants ainsi que l'appauvrissement des caractéristiques génétiques de l'ensemble de la population.

L'ensemble des possibilités de modulation détaillé au paragraphe 2.6.9 a été testé. La modulation finalement retenue consiste à calculer l'intervalle de variation des fonctions d'adaptation des individus constituant la génération courante ainsi que leur moyenne. Puis d'appliquer une transformation linéaire visant à fixer un nouvel intervalle de variation constant de générations en générations.

Cette approche permet d'assurer un meilleur parcours de l'espace des solutions en favorisant de manière mesurée les super individus.

Celle-ci a été retenue car elle donne de bons résultats avec un facteur de modulation fixé à deux et ce comparativement à la troncature sigma, qui elle est beaucoup plus coûteuse en temps de calcul.

La figure ci-dessous illustre la modulation retenue.

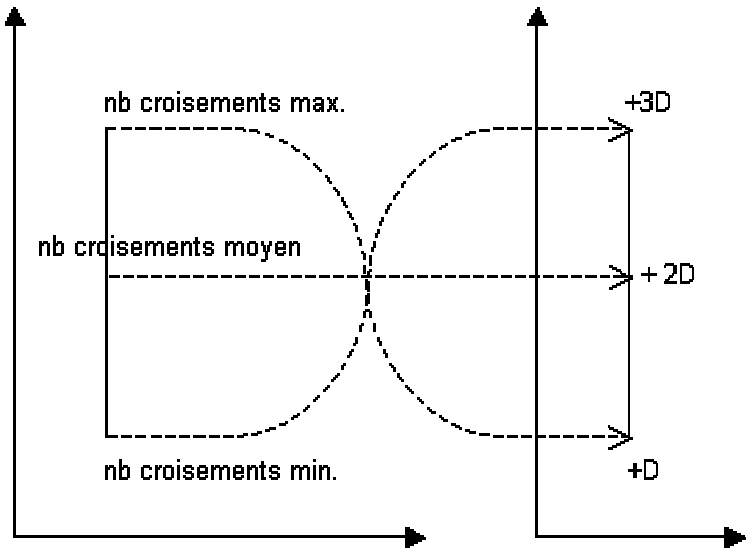


Figure 3-15 : Opérateur de reproduction.

3.3.4.2.6 Maintien de l'élite

Cette variante introduite aux algorithmes génétiques permet le maintien systématique vers la génération suivante de quelques individus les plus performants. Elle a été implémentée dans la chaîne de traitement.

L'expérimentation a démontré que non seulement ce traitement était relativement peu coûteux en temps de calcul, mais de plus qu'il compensait l'effet néfaste combiné des opérateurs de reproduction et de modulation de l'adaptation sur la disparition éventuelle de super individus.

La taille respectable de la dimension des générations traitées à autorisé le maintien de plusieurs super individus par génération.

Le principe retenu a été de maintenir systématiquement par recopie, les dix meilleurs graphes de la génération courante, en évitant la recopie de doublons.

L'expérimentation n'a pas réussit à mettre en évidence la nécessité ou non de prendre en compte la recopie de doublons, celle-ci n'a pas été maintenue simplement sur la base de considérations théoriques, l'objectif étant de limiter l'appauvrissement génétique de la population des graphes traités.

3.3.4.2.7 Traitement des parents

L'opérateur de croisement appareille les chaînes retenues pour participer à la création d'une nouvelle génération.

Un traitement supplémentaire à été introduit ici, il tente d'éviter la proximité des individus identiques dans la génération courante.

En effet croiser des individus parents identiques ne peut produire que des individus enfants identiques aux parents, ce qui limite fortement l'intérêt de l'opérateur de croisement.

De plus il est important de signaler que dans le cadre du maintien de l'élite, les individus sont classés par ordre décroissant de leur adaptation ce qui peut rapprocher les individus identiques.

Ce traitement complémentaire, peu coûteux en temps de traitement, va dans un premier temps redistribuer de façon aléatoire les différents individus dans la génération courante, puis il va contrôler durant une deuxième passe la non proximité d'individus similaires et tenter de les séparer.

Il est à noter que le critère de similarité retenu s'appui sur la fonction d'adaptation des chaînes et non sur la position des sommets, ce qui peut constituer un biais.

3.3.4.2.8 Impact du classement de l'individu originel

Nous avons vu plus haut qu'il était important de maintenir les algorithmes déterministes d'identification des isthmes et points d'articulation en vu d'améliorer l'organisation de l'individu originel exploité par l'algorithme génétique.

L'expérimentation a permis de confirmer cela.

Des tests ont été effectués à l'aide du graphe de référence numéro trois.

La moyenne des nombres de croisements observés suite à l'application de l'algorithme génétique seul (200 runs) a été comparée à la moyenne des nombres de croisement observés suite à l'application d'une chaîne de traitement composée cette fois du même algorithme génétique appliqué suite à un pré-traitement déterministe intégrant l'identification des isthmes et des points d'articulation, triant les sommets et les composantes connexes dans la chaîne de caractères abritant les sommets du graphe.

3.3.4.2.9 Dimension des générations

Le nombre de deux cents individus semble dans le cadre de notre problématique un bon compromis. Non seulement les temps de calcul engendrés ne sont pas excessifs mais de plus la dimension probabiliste de l'opérateur de reproduction peu s'exprimer.

Il est important de noter que la dimension des générations n'a pas d'impact sur la capacité de l'algorithme génétique à optimiser un graphe de référence plutôt qu'un autre, seuls les temps de calcul sont impactés et ce de manière très significative.

Des tests ont été effectués dans ce sens, les moyennes des critères d'esthétisme calculées sur la base de deux cents runs par graphe de référence et ce pour des générations de dimension successivement fixées à 100, 200, 400 et 500 individus ont mis en évidence une quasi stabilité des adaptations des individus les plus performants dès que le nombre de 200 individus par génération a été atteint.

3.3.4.2.10 L'opérateur de mutation

Durant cette phase d'expérimentation, il a été consacré un temps très important à la mise en évidence de l'utilité de cet opérateur.

En effet cet opérateur est sensé compenser la perte de patrimoine génétique engendrée par les autres opérateurs mis en œuvre.

Le résultat est plutôt décevant. En effet pour l'ensemble des graphes de référence ainsi que pour de nombreuses combinaisons de ceux-ci, les moyennes des adaptations des individus les plus performants retenus suite à l'exécution de deux cents runs successifs ont donné des résultats comparables, que l'opérateur de mutation soit appliqué ou non.

Malgré l'augmentation importante des temps de calcul observée (en moyenne 7%) cet opérateur a été maintenu.

3.3.4.3 Paramètres retenus

3.3.4.3.1 Expérimentation

Sont listés ci-dessous les principaux paramètres déterminés durant la phase d'expérimentation. La liste fournie ici n'est pas exhaustive, certaines adaptations de l'algorithme génétique ayant été retenues soit parce qu'elles étaient fortement recommandées par la littérature, soit parce que pour certaines conformations de graphes elles se sont avérées incontournables.

- Taux de sélection = 0,95 ;
- Taux de reproduction par crossing-over = 1 ;
- Taux de mutation = 0,001 ;
- Taille d'une population = 200 ;
- Nombre d'individus élitistes conservés = 10 ;

Cette combinaison de paramètres semble être un bon compromis, l'algorithme correspondant a été appliqué à l'ensemble des graphes de référence ainsi qu'à des combinaisons de graphes de référence et ce avec succès.

3.3.4.3.2 Conditions d'arrêt

La condition d'arrêt retenue est similaire à celle implémentée dans l'algorithme de recuit simulé.

Il s'agit de la combinaison de deux critères, l'absence sur les trois dernières générations d'amélioration du critère d'esthétisme, ainsi que la définition d'une durée maximale d'exécution de la chaîne de traitement.

Les tests effectués sur l'ensemble des graphes de référence ont mis en évidence que si aucune amélioration n'été apportée sur le graphe en cours d'optimisation durant les trois dernières générations, aucune amélioration ne serait apportée sur les sept génération à venir.

Quelque soit la condition d'arrêt ayant entraîné l'arrêt de l'algorithme, la meilleure des conformations du graphe obtenue durant l'optimisation par l'algorithme génétique sera retenue.

3.3.4.4 Résultats observés

3.3.4.4.1 Apport de l'algorithme génétique

La courbe de réponse de l'algorithme de recuit simulé illustre sa dynamique. En effet le graphe de référence ainsi traité voit son nombre de croisements chuter rapidement, pour ensuite se stabiliser.

Ceci est essentiellement dû au paramétrage de la condition d'arrêt de l'algorithme de recuit simulé retenu comme cela est détaillé au paragraphe 3.4.3.1.4. En effet notre objectif n'est pas d'effectuer une optimisation complète des graphes par le biais de cet algorithme, mais seulement de commencer à l'organiser, pour soulager l'algorithme génétique ensuite appliqué.

Ceci est aussi dû aux caractéristiques intrinsèques de l'algorithme de recuit simulé, qui comme le démontre la figure suivante, est plus adapté pour pré-traiter les graphes que pour les optimiser de manière fine.

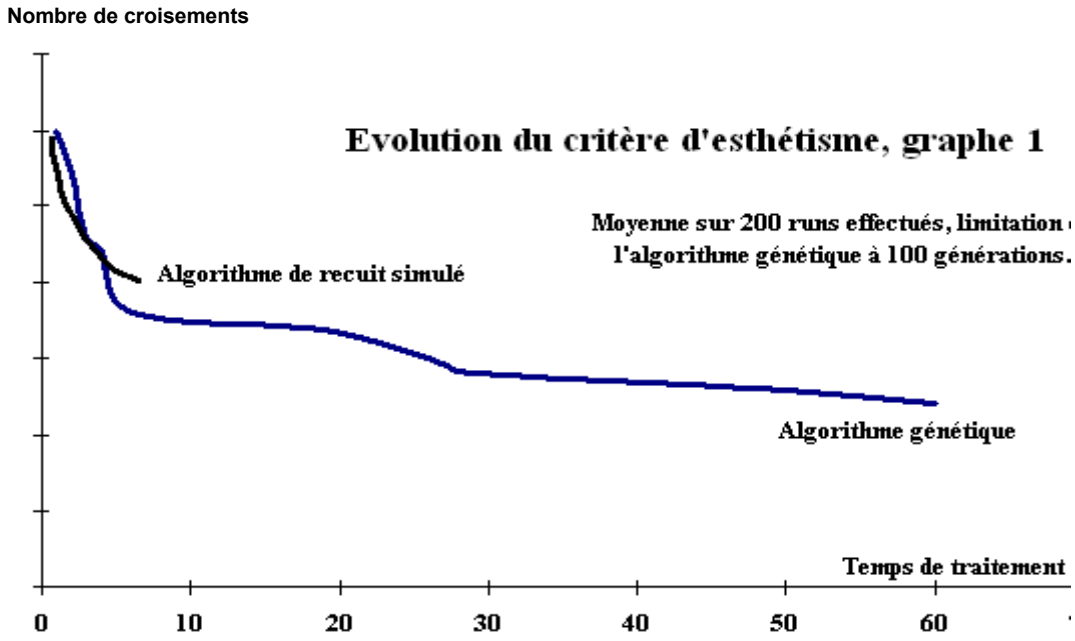


Figure 3-16 : Complémentarité des algorithmes stochastiques.

3.3.4.4.2 Pré-positionnement

Les temps de traitement ont été très nettement améliorés par le pré-positionnement des sommets des graphes à l'aide de l'algorithme de recuit simulé.

En effet nous avons évoqué au paragraphe 2.6.8 la notion de schème et au paragraphe 3.5.2.8 les effets de l'identification des isthmes et points d'articulation sur l'amélioration de la convergence de l'algorithme génétique.

Le recuit simulé participe à cette tâche, dans la mesure où ne pouvant influencer l'ordonnancement des sommets dans la chaîne de codage, l'optimisation qu'il va apporter va conduire à l'identification des schèmes par ajustement des positions des sommets. La figure 3.17 illustre ce concept.

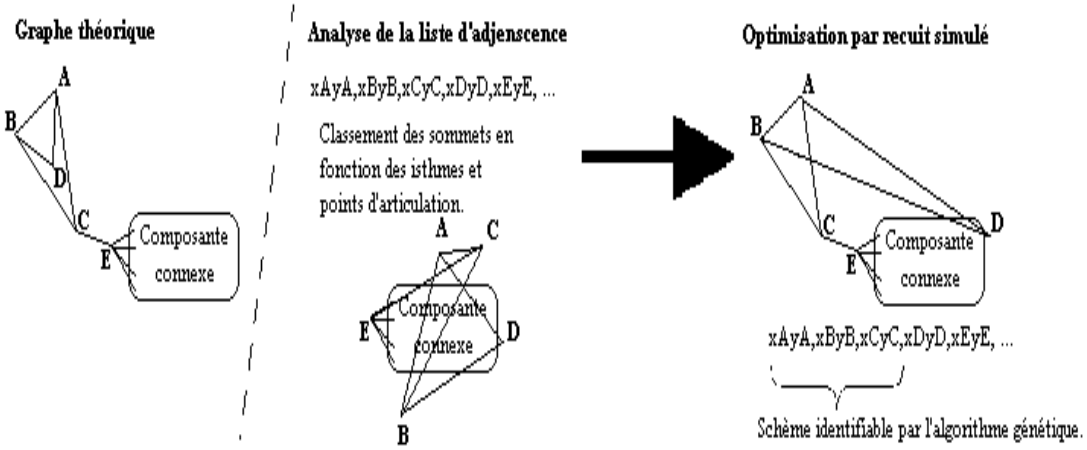


Figure 3-17 : Intérêt du pré-positionnement du graphe avant application de l'algorithme génétique.

Le recuit simulé favorise donc l'identification de schèmes initialement créés par les algorithmes déterministes, ces schèmes seront ensuite maintenus et enrichis par l'algorithme génétique.

Ceci participe aussi à justifier la mise en œuvre d'une structure de codification du graphe à optimiser commune à l'ensemble de la chaîne de traitement.

4 APPLICATION A LA BIBLIOMETRIE

Cette application s'appuie sur les données produites dans le cadre d'une analyse confiée en 1995 au laboratoire du Centre de Recherche Rétrospective de Marseille.

Le détail de cette étude est accessible via l'article écrit par Boutin et *ali* [Boutin-96], qui décrit une exploitation de l'approche réseau dans le domaine de la bibliométrie.

Ce type d'application, qui a fait l'objet de nombreux travaux, consiste à mettre en évidence les structures de relations qui existent entre différentes entités notamment l'activité de co-citation, les réseaux socio-techniques, la structure thématique d'un domaine ou encore les collaborations scientifiques.

La démarche mise en oeuvre s'appuie sur l'analyse de références bibliographiques téléchargées dont le nombre peut nécessiter l'exploitation d'outils spécifiques permettant de mieux appréhender l'information sous-jacente.

La procédure suivie implique plusieurs entités de traitement conduisant à la construction automatique d'un réseau de relations entre les différentes unités bibliographiques, réseau indispensable à la bonne compréhension du phénomène étudié dès que le nombre de références analysées devient important.

L'analyse bibliométrique effectuée pour le compte du laboratoire du C.R.R.M. en 1995 a impliqué la construction du réseau des auteurs travaillant dans le domaine de la bibliométrie au niveau mondial pour tenter de répondre aux questions suivantes :

- Quelles sont les différentes équipes de recherche travaillant sur ce sujet ?
- Quelle est la dynamique de ces différentes équipes ?

La procédure suivi a été constituée de trois étapes couramment employées dans le domaine de la bibliométrie :

- La collecte des données, à savoir le téléchargement de références bibliographiques depuis le Cédérom Pascal 1984-1994. La requête exploitée pour cela a été : "bibliometr? or scientometr? or informetr?", elle a produit un volume de 1191 références, cette information massive n'étant bien sûr pas exploitable par lecture directe.

- Le traitement des références bibliographiques a été réalisé à l'aide du produit Dataview développé au sein du laboratoire du C.R.R.M., et a abouti à la création des matrices exploitées pour construire les réseaux analysés.
- Enfin une analyse réseau a été menée, elle s'est appuyée sur l'exploitation du produit « Matrism » lui aussi développé au sein du C.R.R.M. et est décrite dans les travaux de recherche de M. Boutin [BOUTIN-99].

L'analyse bibliométrique complète effectuée dans le cadre de cette étude n'est pas reprise ici, elle n'offre que peu d'intérêt dans le cadre de ce travail de recherche, seul le premier réseau créé va être exploité comme base d'expérimentation.

La figure 3.18 représente le premier réseau construit dans le cadre de l'étude citée précédemment, les auteurs ainsi que le nombre d'articles qu'ils ont rédigés constituent les sommets du réseau exprimés sous forme de zones de texte.

Les filtres appliqués ont conduit à ne retenir que les auteurs ayant une fréquence de publication supérieure ou égale à 3 et une fréquence de co-publication d'au moins deux articles. Seuls les groupes constitués de plus de trois auteurs ont finalement été retenus.

Six groupes d'auteurs ont ainsi été mis en évidence, la représentation du réseau ainsi constitué est fidèle au rapport édité dans le cadre de l'étude datant de 1995. Son excellente lisibilité s'explique par l'application d'une étape manuelle de positionnement spatial des différents sommets.

Cette étape a conduit à mettre en évidence les six groupes d'auteurs en les représentant dans des espaces de tracé distincts.

De plus il est important de noter que chaque groupe d'auteurs constituant un sous-réseau a été optimisé de façon à améliorer encore la lisibilité de la structure interne des groupes. Les nombres de croisements d'arêtes ont ainsi été minimisés.

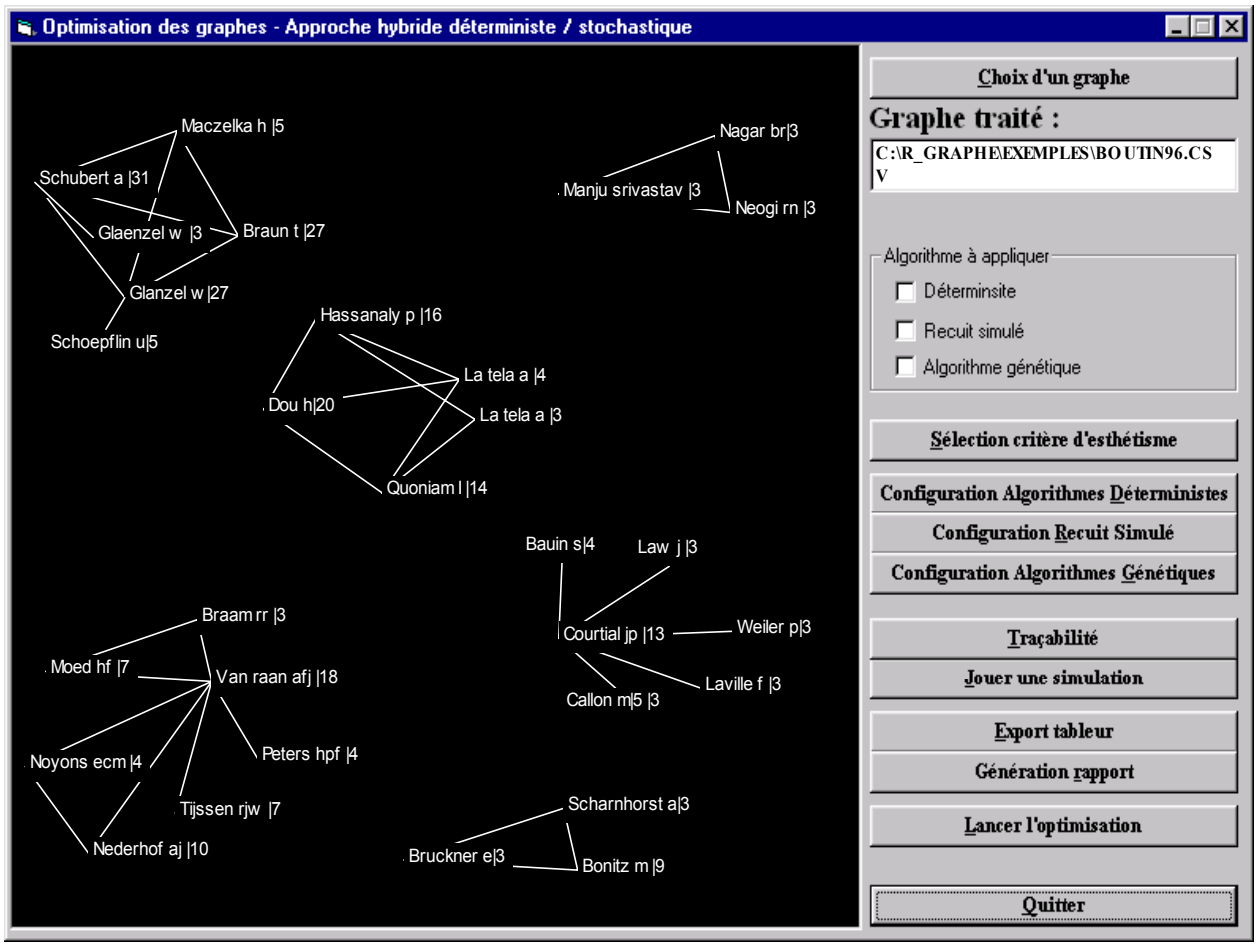


Figure 4-1 : Le réseau d'auteurs.

Les figures qui suivent correspondent à des clichés qui ont été obtenus durant l'application de la chaîne de traitement définie dans le cadre de ce travail de recherche.

Elles sont représentatives des différentes étapes de traitement et contribuent à valider, sur la base d'un exemple réel, les hypothèses émises dans le cadre de ce travail de recherche.

La figure 4-2 représente le premier positionnement aléatoire du réseau exploité, elle n'offre que peu d'intérêt si ce n'est de mettre en évidence la difficulté face à laquelle le spécialiste du traitement de l'information bibliographique peut être confronté lors d'une analyse de réseau s'appuyant sur des outils n'optimisant pas le tracé.

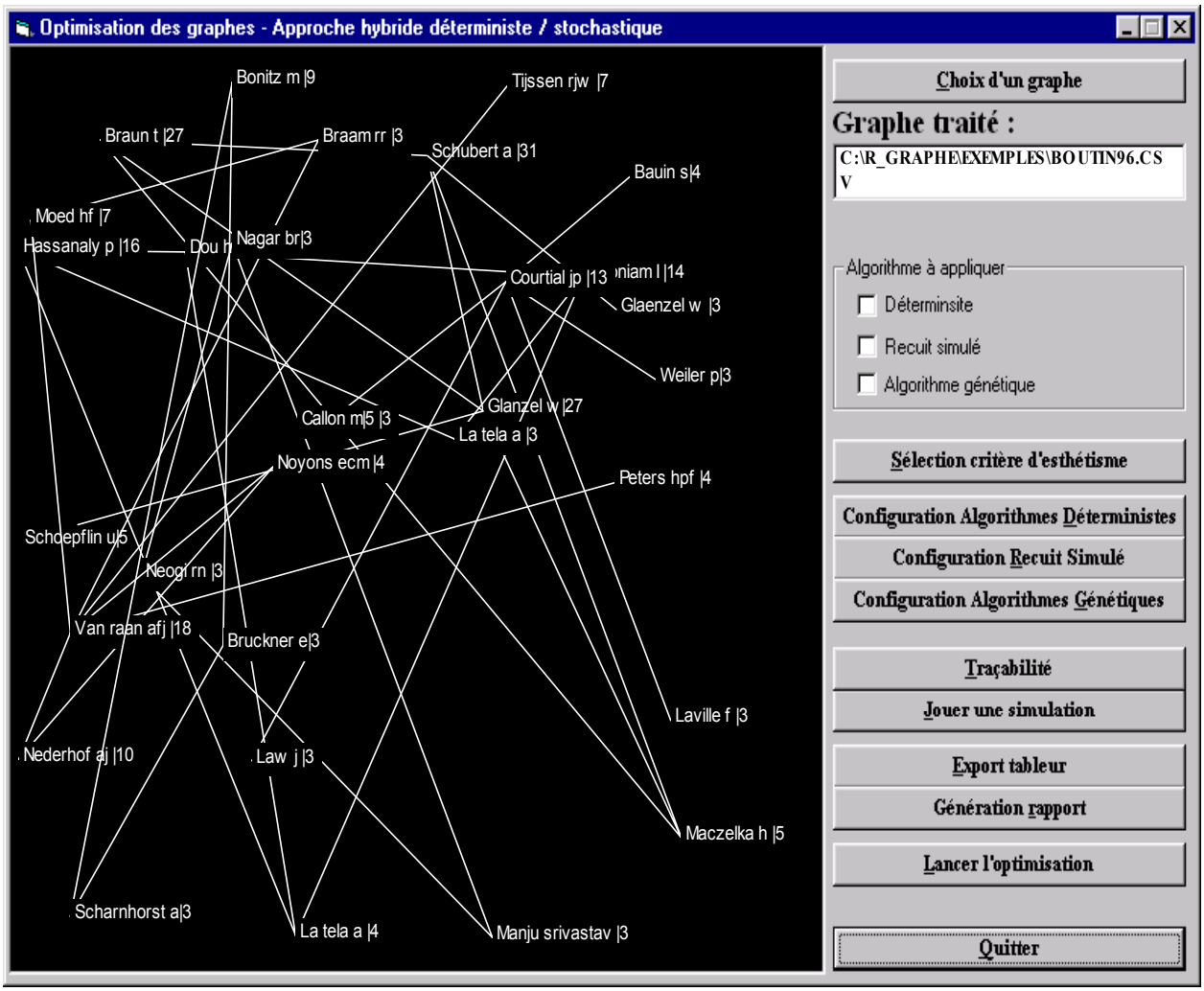


Figure 4-2 : Première étape, le positionnement aléatoire des sommets.

Les différents sommets du réseau ont été légèrement déplacés pour éviter qu'il n'y ait trop de chevauchement d'étiquettes.

Il ne semble pas utile de commenter cette figure, il apparaît clairement que le réseau ainsi constitué n'est pas exploitable, l'utilisateur face à ce type de représentation ressent inévitablement le besoin de déplacer les différents sommets du réseau pour tenter de faire apparaître les structures sous-jacentes.

La première démarche d'une intervention manuelle sera de déplacer les sommets du réseau de manière à rassembler les auteurs liés dans des espaces de tracé distincts.

Ces groupes ainsi constitués seront ensuite retravaillés individuellement de manière à améliorer encore la compréhension par lecture directe des structures contenues dans le réseau. Cette démarche qui est tout à fait instinctive semble triviale, elle nécessite pourtant de la part de l'utilisateur un temps de traitement

manuel non négligeable que l'outil informatique peut aider à réduire, permettant ainsi à l'expert de se concentrer sur sa tache à savoir l'analyse du réseau proprement dite.

La durée de ces manipulations devient considérable si l'on considère que certaines analyses bibliométriques impliquent l'analyse d'un grand nombre de réseaux, ce qui est notamment le cas des analyses dynamiques de réseaux qui mettent en œuvre un grand nombre de représentations chronologiques d'un même réseau.

Les trois figures suivantes ont été produites par l'application des différentes composantes de la chaîne de traitement automatique définie dans le cadre de ce travail, elles contribuent à en illustrer la puissance.

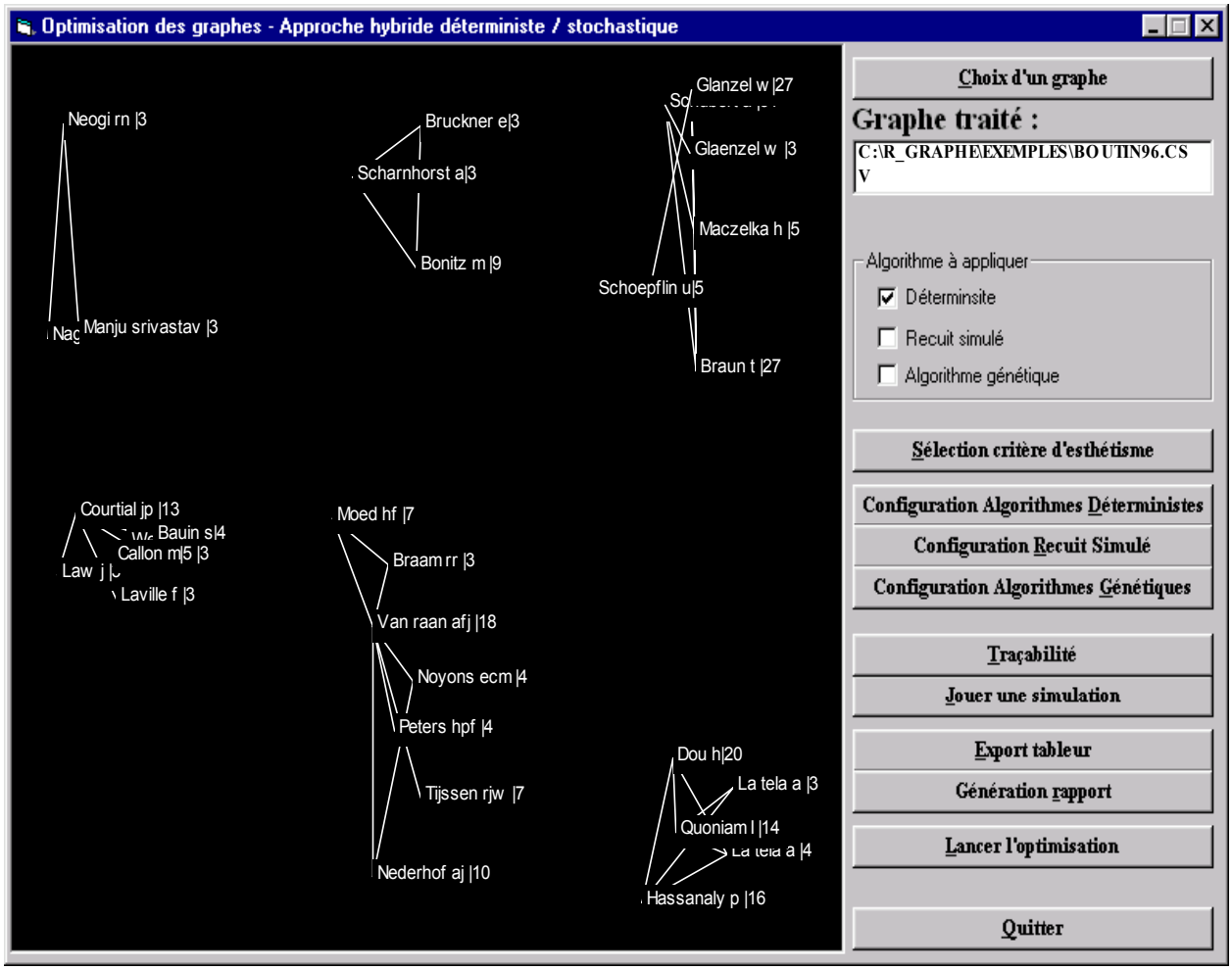


Figure 4-3 : Application de la composante déterministe.

Le processus d'identification des sous-réseaux par application d'algorithmes déterministes illustré par la figure 4-3 est essentiel, son application permet de d'obtenir quasi instantanément une première organisation du réseau.

Les sous-réseaux ainsi identifiés sont exprimés dans des espaces de tracé distincts, les six structures ont été correctement dissociées et ce par application d'algorithmes relativement simples et peu coûteux en temps de traitement.

Les bénéfices apportés par ce type de traitement sont évidents, l'expert dispose ainsi d'une première vision de la structure d'informations qu'il va devoir exploiter, s'il ne peut pas encore répondre à la question « Quelles sont les différentes équipes de recherche travaillant sur ce sujet ?» il peut néanmoins déjà répondre à la question «Combien d'équipes de recherche différentes travaillent sur ce sujet » et ce par application de traitements entièrement automatiques n'impliquant aucune intervention manuelle.

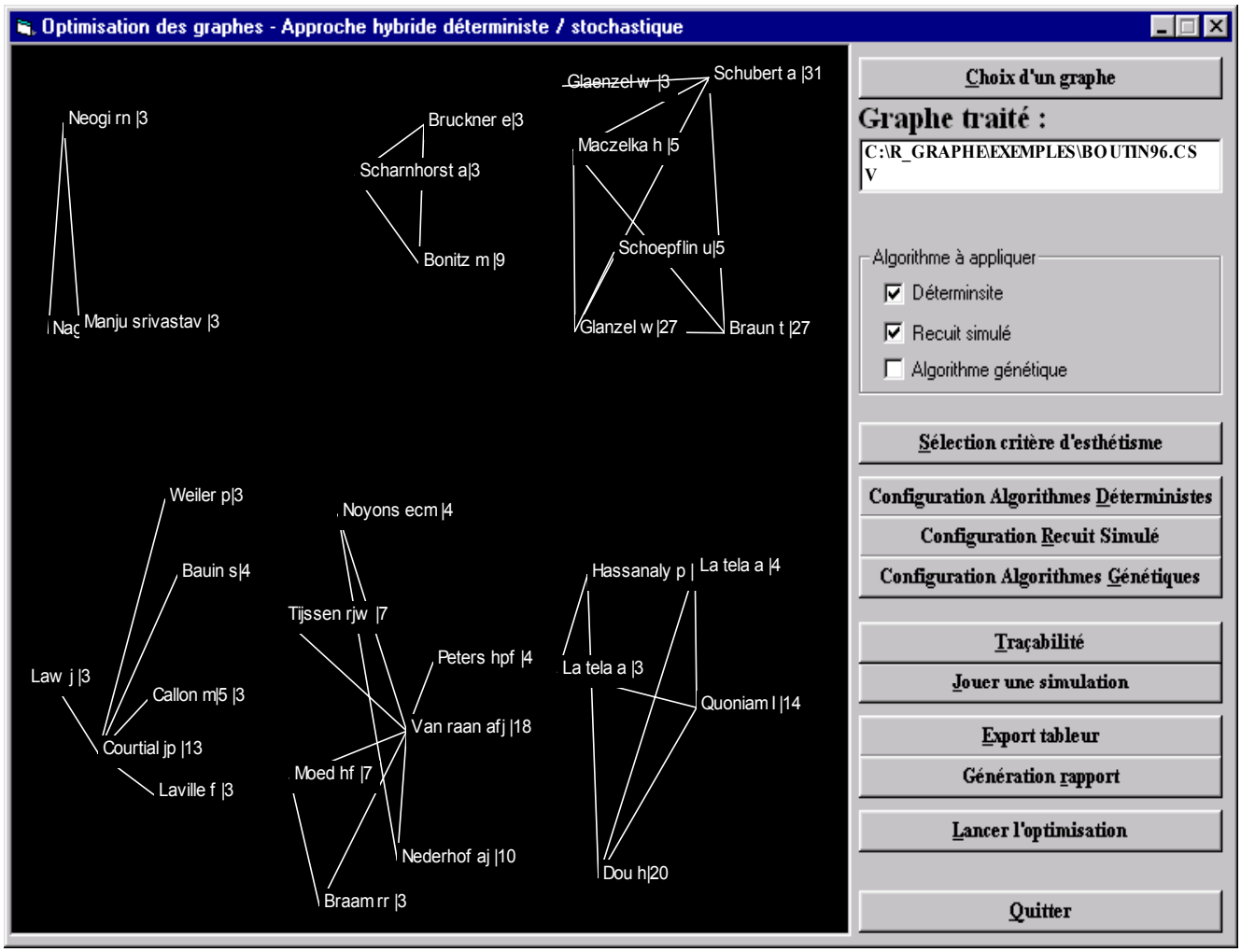


Figure 4-4 : Application de la composante recuit simulé.

La figure 4-4 illustre l'application de la composante « recuit simulé », cette composante contribue à l'optimisation individuelle de chacun des sous-réseaux identifiés précédemment.

Les bénéfices apportés par ce type de traitement sont tout aussi évidents, les sous-réseaux ainsi optimisés voient le nombre de croisements de leurs arêtes diminuer. Les sous-réseaux commencent à s'organiser, l'expert dispose déjà d'une vision suffisamment claire de la structure de chacun des sous-réseaux lui permettant de commencer à répondre à la question : « Quelles sont les différentes équipes de recherche travaillant sur ce sujet ?».

Les traitements mis en œuvre sont ici aussi totalement automatiques, aucune intervention manuelle n'est nécessaire.

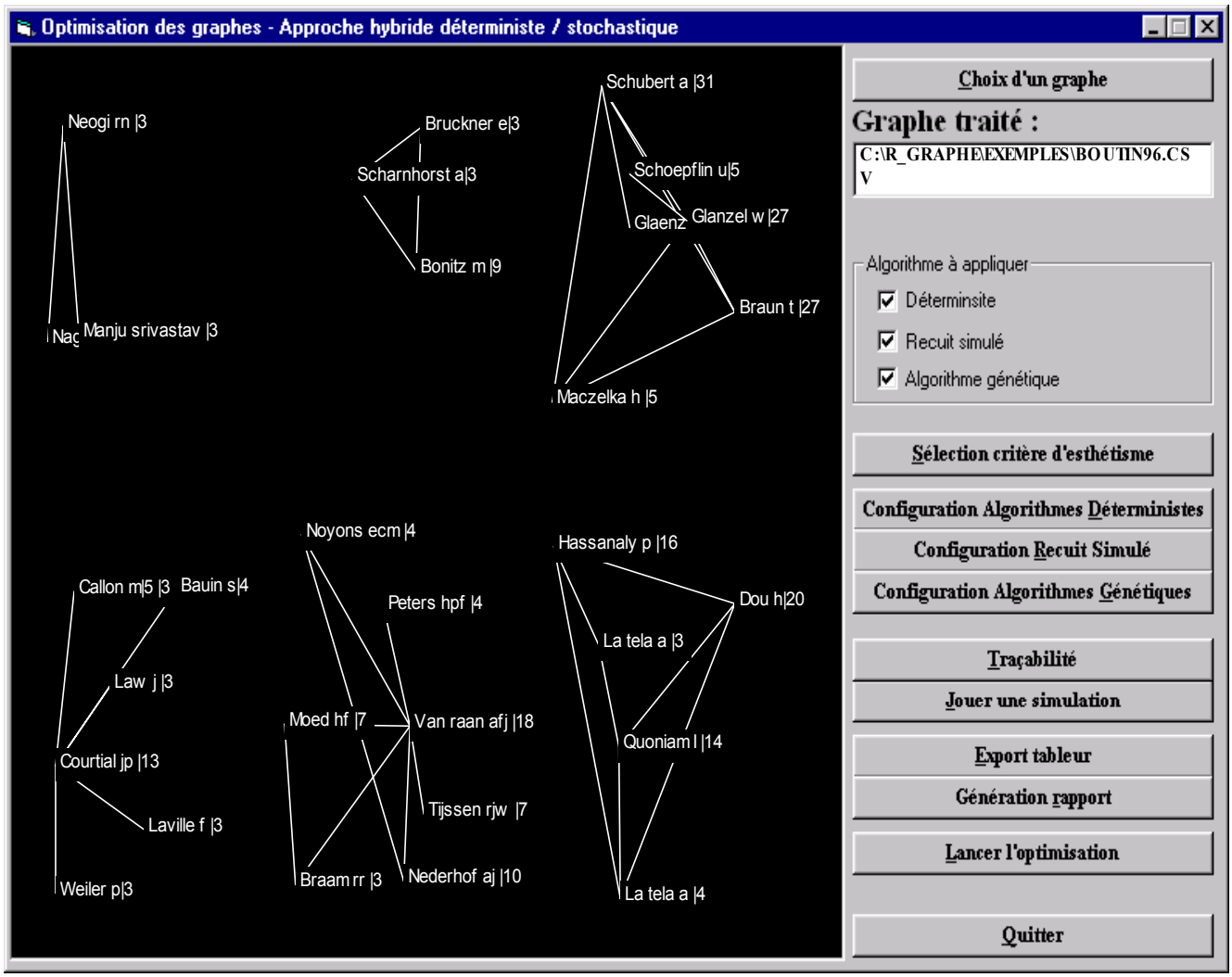


Figure 4-5 : Application de la composante algorithme génétique.

Enfin l'application de la dernière composante, l'algorithme génétique, permet d'améliorer encore l'organisation de certains sous-réseaux en poursuivant les améliorations déjà produites par l'étape « recuit simulé ».

Il est notamment à remarquer la diminution du nombre de croisements du dernier sous-réseau.

Le réseau produit par des traitements totalement automatiques, et qui dans le cas présent ont eu un coût en temps de traitement inférieur à deux minutes sur un ordinateur de type PII 200 Mhz, est déjà susceptible d'être exploité par l'expert.

L'unique retouche manuelle qui pourra alors être envisagée va concerner éventuellement le chevauchement des étiquettes ainsi que la répartition dans l'espace de tracé des arêtes des sous-réseaux, sachant que le critère d'esthétisme retenu ici ne prend en compte que le nombre de croisements, deux arêtes quasi parallèles mais ne se croisant pas sont considérées comme optimisées.

5 CONCLUSION

5.1 CHAINE DE TRAITEMENT « DEFINITIVE »

Une première version de la chaîne de traitement développée dans le cadre de ce travail a déjà été implémentée dans la version 2.b du produit « MATRISME » [Boutin-99] spécialisé en traitement des graphes. Ce produit, très complet, comprend déjà de nombreuses fonctionnalités, notamment des algorithmes de filtrage.

Une version définitive, optimisée, sera implémentée en fin d'année 2002, les améliorations apportées sont relatives au temps de traitement, à l'ajout de l'approche déterministe issue de la théorie des graphes et à l'amélioration de la convergence des algorithmes génétiques.

Certains traitements d'homogénéisation des espaces de tracé, mis en œuvre séparément dans les divers algorithmes tout au long de la chaîne, ont pu être supprimés. D'une part, il y avait redondance, d'autre part, le choix d'une nouvelle codification des individus traités par les algorithmes génétiques nous affranchit de cette problématique. Les temps de calcul ont ainsi été encore améliorés.

De plus, l'analyse d'un critère de qualité du graphe a été implémentée en fin d'application de la composante déterministe de la chaîne de traitement ainsi qu'en fin de traitement de la composante « recuit simulé ». Il s'agit d'une approche prédictive. Ce critère de qualité va permettre d'éviter l'application de l'algorithme de recuit simulé ou de l'algorithme génétique si cela n'est pas nécessaire. Il s'agit d'une optimisation dans la mesure où ce ne sera plus l'algorithme génétique qui décidera par lui-même de s'interrompre suite à l'observation de l'absence d'amélioration de la conformation du graphe en cours d'optimisation. Nous évitons ainsi d'appliquer un algorithme qui ne serait pas efficace.

Comme le montrent les illustrations du chapitre 4, nous avons observé, suite à la phase d'expérimentation et de mise au point de la chaîne de traitement, que l'objectif « nombre de croisements minimal » était globalement atteint.

5.2 LISIBILITE D'UN GRAPHE, UNE NOTION SUBJECTIVE

5.2.1 Touche personnelle

Le critère d'esthétisme retenu s'est limité à évaluer le nombre de croisements d'arêtes contenues dans un graphe.

Nous avons pu observer que les illustrations fournies dans ce document sont constituées de graphes dont la répartition des sommets est très équilibrée.

En fait, dans un but pédagogique**,** les sommets des graphes ont été très légèrement déplacés, en prenant soin de ne pas modifier le nombre de croisements obtenus par les traitements automatiques.

Comme cela est détaillé plus haut, la chaîne de traitement combine des algorithmes déterministes ainsi que des algorithmes probabilistes, qui eux ont en charge le traitement des sous-graphes.

Une simple lecture des algorithmes probabilistes développés met en évidence la dissociation qui existe entre le corps de l'algorithme de traitement et l'évaluation du critère à optimiser. Ceci permet donc une grande liberté dans le choix du critère d'esthétisme.

Le nombre de croisements d'arêtes dans un graphe peut tout à fait être complété par la prise en compte des distances inter sommets, par le chevauchement des zones de texte associées aux sommets, par l'homogénéité du remplissage de la zone de tracé, par une taille des sommets variable, etc…

Néanmoins la richesse du critère à optimiser impacte proportionnellement sur les temps de calcul, c'est pourquoi nous avons simplifié celui-ci, la chaîne de traitement créée étant générique.

5.2.2 Adaptation de la chaîne de traitement aux caractéristiques propres à chaque individus

Certains critères d'esthétisme pourront donc être sciemment intégrés ou non dans le calcul de la fonction de coût du graphe à optimiser.

Le choix de retenir certains critères plutôt que d'autres sera fonction des objectifs poursuivis par l'intervenant, ainsi que du compromis qualité de la représentation / temps de calcul, mais le matériel évolue ...

Si l’effet de certains de ces critères est facilement quantifiable et interprétable par l’intervenant, il n’en va pas de même pour une combinaison linéaire de ceux-ci, ainsi que pour l’intégration de critères plus subjectifs.

Comme nous l’avons vu précédemment, la lisibilité d’un graphe est une problématique regroupant des critères objectifs mais aussi subjectifs.

Chaque utilisateur de par son vécu et ses aptitudes personnelles aura plus de facilité à lire un même graphe représenté sous une certaine conformation plutôt qu’une autre.

Il semble difficile d’appliquer une approche analytique pour choisir la configuration des paramètres de la chaîne de traitement en intégrant les habitudes et les besoins particuliers des utilisateurs finaux.

Il semblerait plus naturel d’exploiter un outil permettant à un utilisateur donné de lancer des optimisations sur un panel de graphes prédéfini. Une notation subjective de la clarté des conformations des graphes tracés, permettant alors de déterminer les paramètres de la chaîne de traitement.

Des algorithmes d’apprentissage, comme les algorithmes génétiques ou mieux encore des réseaux de neurones sont tout à fait à même de résoudre cette problématique.

Ils permettraient de définir, après une phase d’apprentissage, les valeurs des paramètres à appliquer à la chaîne de traitement de façon à répondre au mieux aux attentes des utilisateurs. Et ce sans que l’utilisateur ne soit conscient de l’existence de ces paramètres, qu’ils participent à la définition de la fonction coût ou plus généralement au paramétrage de la chaîne de traitement.

6 BIBLIOGRAPHIE

Théorie et utilisation des graphes :

AFNOR-84

AFNOR, *Traitement des résultats de mesure.* Détermination de l'incertitude associée au résultat final, NF X 06-044(1984).

Alaoui-93

Moshine Alaoui, *Reconstruction tridimensionnelle d'images à partir d'un faible nombre de projections par la méthode du recuit simulé*, Thèse INSA-Lyon, 1993.

Aubry-93

Christophe Aubry, *Tracé automatique de graphes*, Document IBM centre européen de Mathématiques Appliquées, 1993.

Berge-83

C. Berge, *Graphes,* Gauthier-villar, 1983.

Bourret-91

P. Bourret, M. Samuelides, *Réseaux neuronaux : Une approche connexioniste de l'intelligence artificielle*, Teknea, 1991.

Celeux-89

G. Celeux, *Classification automatique des données*, Dunod, 1989.

Dalud-94

M. Dalud, *Modèle prétopologique pour une méthodologie d'analyse de réseaux : concepts et algorithmes*, Thèse Lyon 1, 1994.

Davis-91

Lauwrence Davis, *Handbook of genetic algorithms*, Van Nostra, 1991.

Delahaye-95

D. Delahaye*, Optimisation de la sectorisation de l'espace aérien par algorithme génétique,* Thèse de l'école nationale supérieure de l'aéronautique et de l'espace, 1995.

Djouadi-96

Y. Djouadi, *Logique possibiliste et amélioration génétique pour la sélection et l'agencement d'objets cartographiques*, Thèse Lyon I, 1996.

Droesbeke-87

F. Droesbeke, M. Hallin, Cl. Lefevre, *Les graphes par l'exemple*, Ellipses, 1987.

Durand-96

N. Durand, *Optimisation de trajectoires pour la résolution de conflits aériens*, Thèse INP Toulouse, 1996.

Goldberg-91

David Goldberg, *Algorithmes génétiques*, Addison-Wesley, 1991.

Heckenroth -90

Patrick Siarry H. Heckenroth, *Optimisation par la méthode du Recuit Simulé du dessin d'un modèle Conceptuel de Données – AFCET Interfaces*, Décembre 1990.

Hérault-94

J. Hérault, C. Jutten, *Réseaux neuronaux et traitement du signal*, Hermès, 1994.

Holland-75

John Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.

Jodouin-94-1

J.-F. Jodouin, *Les réseaux de neurones, principes et définitions*, Hermès, 1994.

Jodouin-94-2

J.-F. Jodouin, *Les réseaux neuromimétiques*, Hermès, 1994.

Kane-97

C. Kane, *Optimisation topologique des Formes par Algorithmes Génétiques,* Revue Française de mécanique, N°4, P237 :246 1997.

Kirkpatrick-83

S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Optimization by simulated annealing*, Science, Vol 200, N° 4598, p. 671-220, 1983

Lerman-94

I.C. Lerman et R.F. Ngouenet, *Algorithme génétiques séquentiels et parallèles pour une représentation affine des proximités*, Publication interne IRISA-INRIA, Janvier 1994.

Lucas-84

M. Lucas, *Algorithme et représentation des données*, Edition Masson, 1984.

Michalewicz-91

Z. Michalewicz, C. Janikov, *Handling constraints in genetic algorithms*, Proceedings of the fourth Intertational Conference on Genetic algorithm, ICGA, 1991

Ore-70

O. Ore, *Les graphes et leurs applications*, Collection sigma, 1970.

Palmieri-93

X. Qi, F. Palmieri, *The diversification role of crossover in the génétic algorithm*, Proceedings of the fifth Intertational Conference on Genetic algorithm, ICGA, 1993

Pritchard-69

A. Prichard, *Statistical bibliography or bibliométrics*, Journal of publication, vol 25, P348-349, 1969.

Py-90

B. Py, *Statistique descriptive, Nouvelle méthode pour bien comprendre et réussir*, Economica, 1990.

Ranson-94

C. Ranson, Présentation du « *Guide pour l'expression des incertitudes de mesure* », 6ème Congrès International de Métrologie, Lille, 1993.

Rawlins-91

G. rawlins, *Foundations of genetic algorithms*, Morgan Kaufmann, 1991.

Rudolph-94

G. Rudolph, *Convergence Analysis of Canonical Genetic Algorithms*, IEEE Transaction on Neural Networks VOL 5. No1, 1994

Schoenauer-96-1

M. Sebag, M. Schoenauer, *Contrôle d'un algorithme génétique*. Revue d'intelligence artificielle, 1996.

Schoenauer-96-2

C. Kane, M. Schoenauer, *Genetic operators for two-dimensional shape optimization*, Artifical Evolution, Springer Verlag, 1996.

Schomberg-85

R. Schomberg, *Pratiquez l'intelligence artificielle*, Edition Masson, 1985.

Scott-94

J. Scott, *Social Network Analysis*, SAGE Publications, 1994.

Siarry-87

P Siarry, G. Dreyfus, *La méthode du recuit simulé*, I.D.S.E.T., 1987.

Thiria-97

S.Thiria, Y. Lechevallier et al., *Statistiques et méthodes neuronales*, Dunod, 1997.

Vose-91

M. Vose, *Generalizing the notion of schema in genetic algorithm*, Artificial Intelligence, N°50, 1991

Vose-92

A. Nix, M. Vose, *Modeling genetic algorithms with Markov chains*, Annals of Mathematics and Artificial Intelligence, N° 5, 1992

Wasserman-94

S. Wasserman, *Social Network Analysis : Methods and applications*, Cambridge University Press, 1994.

Werra-90

D. Werra, *Eléments de programmation linéaire avec application aux graphes*, Presses polytechniques romandes, 1990.

Les réseaux en science de l'information :

Boutin-96

E. Boutin, P. Dumas, L. Quoniam, H. Rostaing, *Les réseaux comme outil d'analyse en bibliométrie. Un cas d'application : les réseaux d'auteurs*, Les cahiers de la documentation Belge, Vol 50, P 3-13, 1996

Boutin-99

E. Boutin, *Le traitement d'une information massive par l'analyse réseau : méthode, outils et applications*, Thèse Aix-Marseille III, 1999.

Chaumier-90

J. Chaumier, *Analyse et langages documentaires, le traitement linguistique de l'information documentaire*, Entreprise Moderne d'Edition, 1990

Dou-92

H. Dou, *La veille technologique*, Dunod, 1992.

Dou-94

H. Dou, *Méthodologie de la veille technologique,* cours de DEA d'Information Stratégique Veille technologique, Aix-Marseille III, 1994.

Faucompré-97

P. Faucompré, *La mise en correspondance automatique de banques de données bibliographiques scientifiques et techniques à l'aide de la classification internationale des brevets*, Thèse Aix-Marseille III, 1997.

Freeman-94

L.C. Freeman et C.M. Webster, *Interpersonnal proximity in social and cognitive space*, social Cognition 12, 223-247, 1994.

Jakobiak-94

F. Jakobiak, Information, Innovation, Japon, Réseaux, ELF-ATOCHEM, F. Jakobiak 1994

Leitzelman-98

M. Leitzelman, *Mise en place d'un système d'Information stratégiques multicritères facilitant l'intégration des ressources régionales et la prise de décision dans le domaine de l'Environnement, Application à la Ville de Marseille*, Thèse Aix-Marseille III, 1998.

Nivol-93

W. Nivol, *Systèmes de surveillance systématique pour le management stratégique de l'entreprise*, Thèse Aix-Marseille III, 1993.

Retourna-95

C. Retourna, *Analyse de cas concrets d'innovations dans les PME/PMI problématiques et discussions*, Thèse Aix-Marseille III, 1995.

Rostaing-93

H. Rostaing, *Veille technologique et Bibliométrie Concepts, Outils et Applications*, Thèse Aix-Marseille III, 1993.

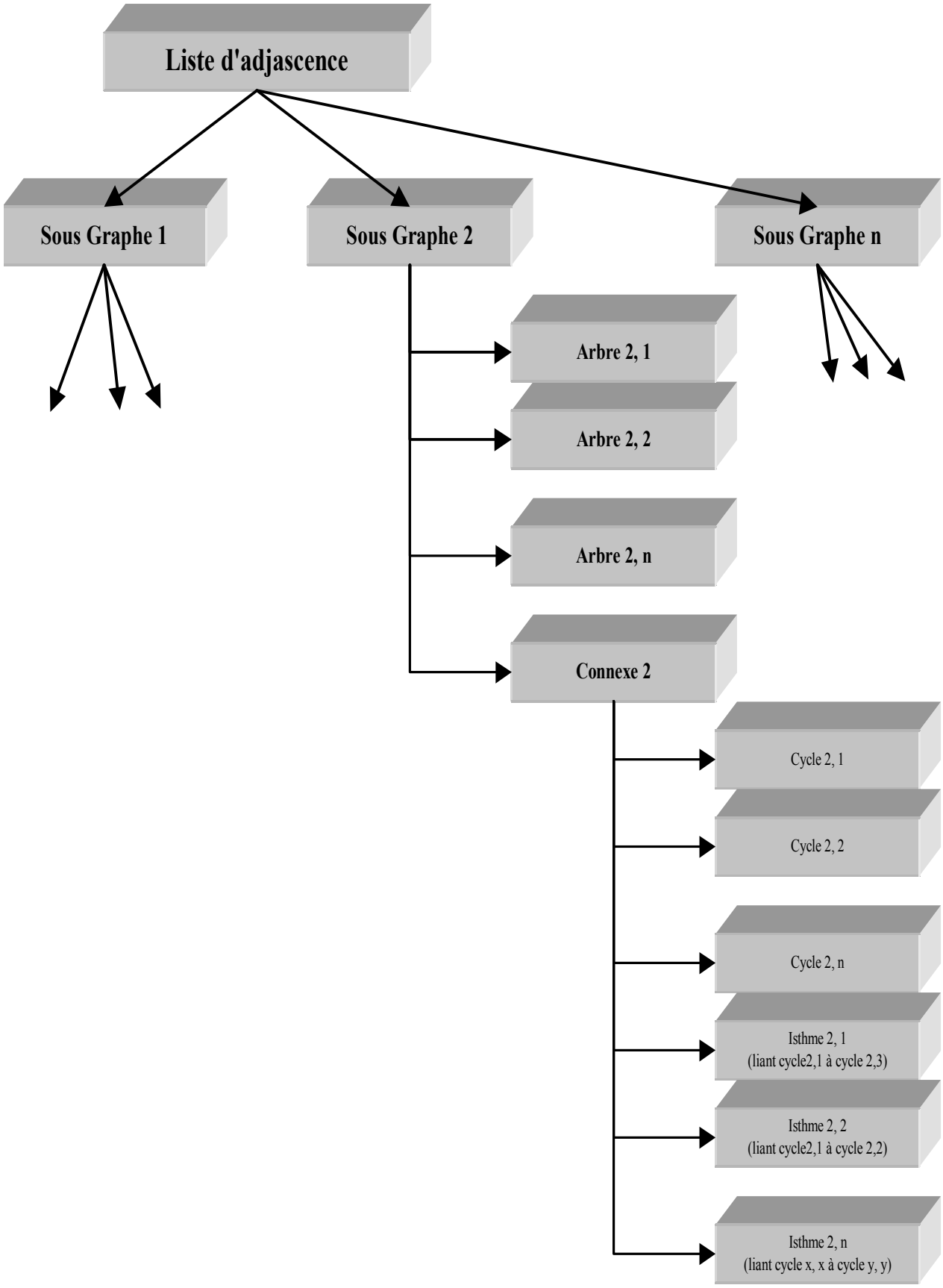
Quelques sites web visités durant l'établissement de la bibliographie :

Sites Web traités	Année de visite
www.aic.nrl.navy.mil	2000
www.ices.fr	2000
www-leibniz.imag.fr	2001
www.mage.univ-mulhouse	2000
csr.uvic.ca	2000
www.cmap.polytechnique.fr	2000
web.inria.fr	2000
www.enst.fr	2001

ANNEXES

ANNEXE 1

Découpage d'un graphe par méthode déterministe

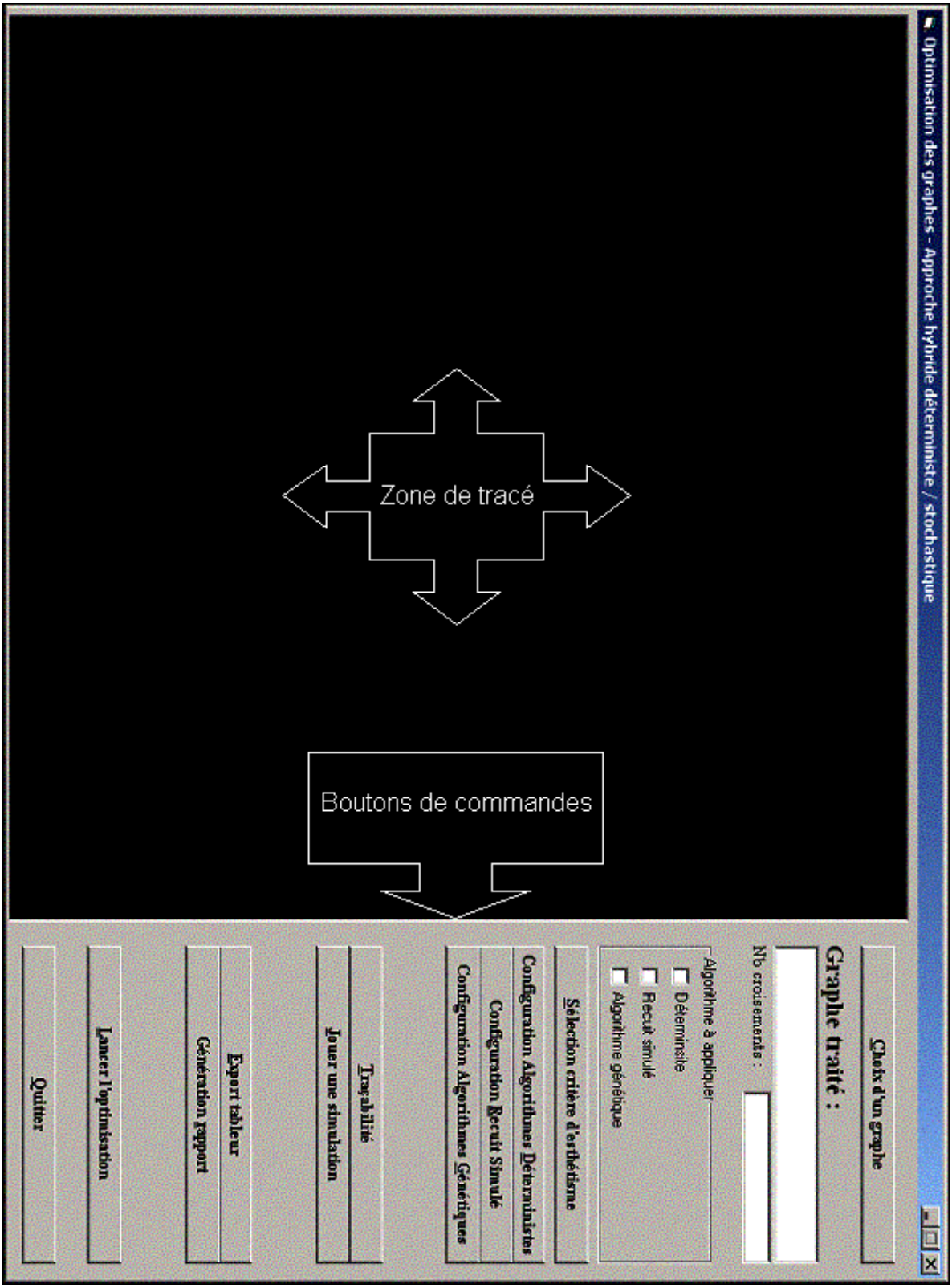

Liste d'adjascence
Sous Graphe 1
Sous Graphe 2
Sous Graphe n
Arbre 2, 1
Arbre 2, 2
Arbre 2, n
Connexe 2
Cycle 2, 1
Cycle 2, 2
Cycle 2, n
Isthme 2, 1
(liant cycle2,1 à cycle 2,3)
Isthme 2, 2
(liant cycle2,1 à cycle 2,2)
Isthme 2, n
(liant cycle x, x à cycle y, y)

ANNEXE 2

Forme principale du programme de test

Nb : cette application a été développée en visual basic version 6.0, elle fonctionne sur systèmes d'exploitation Windows 98, 2000, … en mode 32 bits. Certains des algorithmes testés ont été traduits en visual basic notamment depuis Java et C^{++}.

Aucune bibliothèque scientifique ou mathématique n'a été implémentée dans le cadre de ce travail. L'ensemble des traitements relatifs aux algorithmes mis en œuvre a été développé, testé et mis au point sur la base des informations bibliographiques recueillies.


Optimisation des graphes - Approche hybride déterministe / stochastique
Choix d'un graphe
Graphe traité :
Nb croisements :
Algorithme à appliquer
Déterministe
Recuit simulé
Algorithme génétique
Sélection critère d'esthétisme
Configuration Algorithmes Déterministes
Configuration Recuit Simulé
Configuration Algorithmes Génétiques
Traçabilité
Jouer une simulation
Export tableur
Génération rapport
Lancer l'optimisation
Quitter
Zone de tracé
Boutons de commandes