République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saâd DAHLAB, Blida

N° D'ordre.....

Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Bouhamoum Redouane Chaabane Wail

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie de logiciel

Sujet :

Amélioration de l'analyse visuelle des données par l'ordonnancement des dimensions d'analyse

Soutenu le :

Mme Ben Sekhi	Président
M.lle Reguigue	Examinateur
M.lle S-Bacha	Examinateur
Mlle. N.BENBLIDIA	Promotrice
Mlle. K.AMEUR	Encadrante

Promotion

2013 / 2014

Résumé

L'objectif de l'exploration visuelle des données est de permettre à l'utilisateur d'obtenir un aperçu des données, en tirer des conclusions, et d'interagir directement avec les données à l'aide des techniques de visualisation. Ces techniques fournissent un plus haut degré de confiance dans l'exploration visuelle des données. Une des techniques de visualisation des données multidimensionnelles fréquentes est les coordonnées parallèles. Dans ce mémoire, nous nous sommes intéressés à la façon de visualiser efficacement les clusters et les données multidimensionnelles en coordonnées parallèles dans le but de faciliter la découverte de connaissances. En particulier, nous avons proposé une approche qui détermine efficacement un bon ordre de coordonnées pour réduire l'encombrement visuel et montrant le lien entre les données représentées pour permettre à l'analyste d'observer les modèles pertinents dans les données et ainsi fournir un meilleure, et nettement plus rapide interprétation. Pour résoudre ce problème, nous l'avons relié au problème de chemin. Ce qui nous connecte à différentes solutions efficaces et conduit à des algorithmes très rapides.

ملخص

الغرض من استخراج البيانات البصرية هو السماح للمستخدم الحصول على لمحة عامة عن البيانات واستخلاص النتائج والتفاعل مباشرة مع البيانات باستخدام تقنيات العرض. توفر هذه التقنيات على أعلى درجة من الثقة في استخراج البيانات البصرية. احدى التقنيات المستخدمة لعرض البيانات متعددة الأبعاد هي الاحداثيات الموازية. في هذه المذكرة، ركزنا على كيفية عرض مجموعات البيانات متعددة الابعاد بشكل فعال في الاحداثيات الموازية من اجل تسهيل اكتشاف المعرفة. على وجه الخصوص، نود العثور على ترتيب جيد للإحداثيات بشكل فعال للحد من الفوضى البصرية ولإظهار الرابط بين البيانات الممثلة لكي نسمح للمحلل بمراقبة النماذج ذات صلة في البيانات ولتفسير جيد وسريع. لحل هذه المشكلة، ربطناها مع مشكلة مسار هاميلتون. هذا التعريف يرتبط بمختلف الحلول ذات النتائج الفعالة ويؤدي الي خوارزميات سريعة جدا.

Abstract

The important goal of visual Data Exploration is to allow the user to get an overview of the data, draw conclusions, and interact directly with the data using visualization techniques. These techniques provide much higher degree of confidence in the visual exploration of data.

One of the frequently used to visualize multidimensional data is Parallel Coordinates. In this Master thesis, we are interested in how to effectively visualize clusters and multi-dimensional data in parallel coordinates in the purpose of facilitating knowledge discovery. In particular, we would like to efficiently find a good order of coordinates for reducing visual clutter and showing the link between the represented data to allow the analyst to observe relevant patterns in the data for good and fast interpretation. To solve this problem, we link it to the Hamiltonian path. That connects us to various efficient solutions and leads to very fast algorithms.

Remerciements

Tout d'abord, louange à « Allah » qui nous a guidés sur le droit chemin tout au long du travail et nous a inspirés les bons pas et les justes reflexes. Sans sa miséricorde, ce travail n'aura pas abouti.

Au terme de ce travail, nous tenons à exprimer toute notre reconnaissance et remerciements à notre promotrice Mlle Benblidia Nadjia, dont l'encadrement a été des plus exemplaires et pour la confiance qu'elle nous a attribué. Nos remerciements vont aussi à Mlle Ameur Khadidja qui a été d'un grand apport pour la réalisation de ce travail. Ses conseils ainsi que ses orientations nous ont permis de mener à terme ce projet.

Nous tenons à mentionner le plaisir que nous avons eu à étudier à l'université de Blida. Nous en remercions ici tous les enseignants de la faculté.

Nous n'oublions pas nos parents et toute la famille Hamane spécialement la tante pour leur soutien.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenus et encouragés au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Sommaire

Liste de figures

Liste des tableaux

Introduction générale

Les progrès réalisés dans la technologie du matériel permettent aux systèmes informatiques d'aujourd'hui de stocker de très grandes quantités de données. Des chercheurs de l'Université de Berkeley ont estimé que chaque année, environ 1 Exa-byte (= 1 million de téraoctets) de données sont générées, dont une grande partie est disponible sous forme numérique. Cela signifie que dans les trois prochaines années plus de données seront générés que dans toute l'histoire humaine avant [1]. Habituellement, de nombreux paramètres sont enregistrés, ce qui entraîne des données multidimensionnelles avec une grande dimension. Cependant, Trouver les informations utiles caché en eux est une tâche difficile. Si les données sont présentées textuellement, la quantité de données qui peut être affiché atteint une centaine, cette quantité représente une goutte d'eau lorsqu'il s'agit d'ensembles de données contenant des millions d'éléments de données. N'ayant pas de possibilité d'explorer convenablement les grandes quantités de données qui ont été recueillies en raison de leur utilité potentielle, les données deviennent inutiles.

Pour que le *Datamining* soit efficace, il est important d'inclure l'homme dans le processus d'exploration de données et de combiner la flexibilité, la créativité et les connaissances générales de l'humain avec la capacité de stockage énorme et la puissance de calcul des ordinateurs actuels.

L'analyse visuelle des données vise a touché cet objectif, l'idée de base de l'exploration visuelle des données est de présenter les données sous une forme visuelle, permettant à l'homme d'obtenir un aperçu des données, en tirer des conclusions, et d'interagir directement avec les données. Les techniques visuelles d'exploration de données se sont avérées d'une grande valeur dans l'analyse des données et ils ont aussi un fort potentiel pour explorer de grandes bases de données.

En conséquence, l'exploration visuelle de données permet généralement une exploration de données plus rapide et donne de meilleurs résultats, en particulier dans les cas où les algorithmes automatiques échouent souvent. En outre, les techniques d'exploration de données visuelles parmi lesquelles nous citons la technique des coordonnées parallèles fournissent un degré beaucoup plus élevé de confiance dans les résultats de l'exploration. Ce fait conduit à une forte demande pour des techniques d'exploration visuelle et les rend indispensable en conjonction avec des techniques automatiques d'exploration de données.

- Problématique

Les coordonnées parallèles [2] ont été largement utilisées pour analyser des ensembles de données de grande dimension. En représentant les dimensions comme des axes parallèles, les éléments de données deviennent des poly-lignes en reliant leurs valeurs sur les axes, les coordonnées parallèles peuvent représenter des données à N dimensions dans un espace à deux dimensions. Cette structure permet la visualisation d'un nombre important de données. Cependant, lorsque ce grand nombre est affiché, les coordonnées parallèles peuvent devenir trop denses, la représentation devient encombrée, les poly-lignes croisés et chevauchés et les relations entre les données deviennent cachées ce qui rend l'interprétation difficile ou quasiment impossible. Par conséquent, la réduction de l'encombrement visuel causé par les croisements et chevauchements excessives des lignes et comme la représentation des relations entre les données s'avère très importante pour les coordonnées parallèles afin de comprendre, interpréter et extraire des connaissances, plusieurs techniques de l'amélioration de l'affichage ont été proposées parmi elles, l'ordonnancement des dimensions. De ce fait, notre problématique principale concerne l'encombrement de l'affichage et la représentation des relations entre les données caché par celle-ci, ce qui rend l'interprétation de graphe difficile.

- Objectif

L'ordonnancement des dimensions peut avoir un impact majeur sur l'expressivité de la visualisation. Différents ordres de dimensions peuvent révéler différents aspects des données et affectent l'encombrement perçu et la structure à l'écran, tirant complètement différentes conclusions de chaque ordre. L'ordonnancement manuel des dimensions est disponible dans certains systèmes permettant aux utilisateurs de modifier manuellement l'ordre des dimensions à partir d'une liste de dimensions reconfigurable. Mais, la recherche exhaustive du meilleur ordre est fastidieuse, même pour un petit nombre de dimensions. Conscients de l'importance de l'ordre des dimensions dans une visualisation multidimensionnelle, notre travail a pour objectif d'ordonner de façon automatique les dimensions sur coordonnées parallèles pour permettre une meilleure visualisation des données.

- Organisation

Afin d'atteindre l'objectif cité ci-dessus, notre mémoire s'organisera comme suit :

Chapitre I : Nous y introduirons l'analyse visuelle des données, ces différentes techniques et décrit son processus. Il montrera par la suite son importance et le rôle de l'homme dans le processus de l'analyse visuelle. Il mettra aussi l'accent sur la visualisation des données multidimensionnelles et ses différents types. Après avoir vue les différent techniques de

visualisation il détaille les coordonnées parallèle qui fera l'objet de notre étude, il parle de l'impact d'ordre des dimensions et le relie au problème de voyageur de commerce.

Chapitre II : Nous y parlerons de problème de voyageur de commerce, le définira, montrera son importance dans l'optimisation combinatoire et son application dans différent domaines de la vie. Il s'intéressera aussi à la résolution du problème et les différentes méthodes proposées.

Chapitre III : Nous viserons à fusionner les deux premier chapitre afin d'appliquer les algorithmes de résolution de problème du voyageur de commerce sur les coordonner parallèles. Il Comportera notre processus d'analyse visuelle qui ajoute l'étape de l'amélioration et les prétraitements nécessaires sur les données. Après il décrira l'étape de l'amélioration composées de calcul de la matrice de similarité sur laquelle les algorithmes d'ordonnancement seront appliqué.

Chapitre IV : Il portera sur la réalisation d'une application d'analyse visuelle de données qui propose automatiquement le meilleur ordre des dimensions et une amélioration de la présentation. Cette réalisation sera suivie d'une série de tests pour voir le plus qu'apportera notre approche à la visualisation et à la tâche d'extraction de connaissances. Il présentera aussi notre application intitulé ClusterViz.

Nous terminerons notre mémoire par une conclusion. Dans laquelle, nous citerons quelques perspectives d'amélioration de l'application ClusterViz que nous jugeons utile en termes de compréhension, interprétation des données, et extraction des connaissances.

Chapitre

I. Analyse visuelle des données

I.1 Introduction

Des techniques d'analyses automatiques telles que les statistiques et l'exploration de données ont été développées indépendamment des techniques de visualisation et d'interaction. Cependant, quelques réflexions clés ont changé la portée plutôt limitée des champs dans ce qu'on appelle aujourd'hui la recherche sur l'analyse visuelle. Une des étapes les plus importantes dans cette direction a été la nécessité de passer de l'analyse confirmatoire des données (à l'aide des graphiques et autres représentations visuelles à seulement présenter les résultats) à l'analyse exploratoire de données (qui permet d'interagir avec les données et les résultat), ce qui a été dit la première fois dans la communauté de la recherche statistiques par John W. Tukey en 1977 dans son livre, analyse exploratoire des données [3].

Dans les dernières décennies, plusieurs méthodes d'analyse ont été développées qui ont un caractère purement automatique ou purement visuel, mais pour faire face à la complexité de problème, l'être humain doit être inclus dans processus d'analyse de données. La découverte de connaissances et d'exploration de données (KDD) est semi ou entièrement automatique. Ces méthodes d'analyse automatique font partie d'une discipline d'une longue et solides tradition et fondations théoriques. Ils ne sont pas centrés sur un domaine d'application, et les contributions du celui-ci sont plus sur des méthodes générales. Les méthodes de KDD sont particulièrement adaptées aux problèmes d'analyse dans lesquels il existe des moyens pour évaluer la qualité des solutions proposées. Cependant, très souvent, ils deviennent des méthodes dans les mains des utilisateurs finaux (par exemple, les médecins) ou les algorithmes peuvent fournir des résultats qui ne conduisent pas à une solution au problème, car ils ne prennent pas en compte les connaissances de l'expert. En revanche, les méthodes de visualisation utilisent les connaissances de fond, la créativité et l'intuition pour résoudre le problème. La figure I-1 compare le processus KDD et la visualisation de l'information. L'analyse visuelle apporte au fond les connaissances des experts dans le processus d'analyse, ainsi que la capacité d'interagir et de diriger le processus d'analyse.

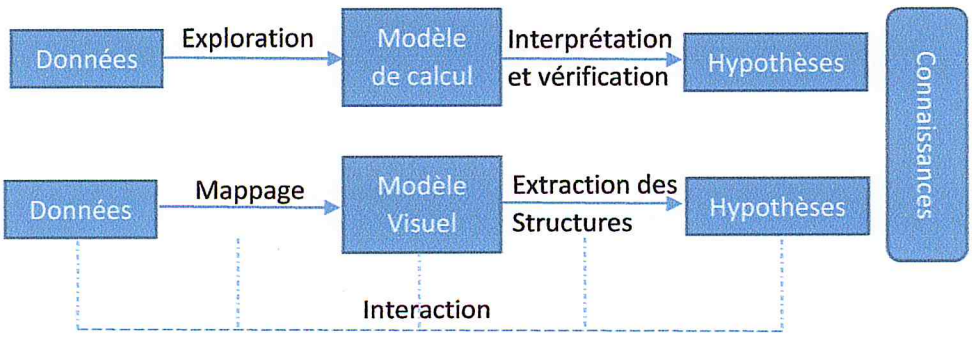


Figure I-1 : Processus de KDD et le processus de l'analyse visuelle [4]

Cette intégration élargi considérablement le champ d'application à la fois la visualisation de l'information et les champs d'exploration de données, résultant en de nouvelles techniques et de nombreuses possibilités de recherche intéressantes et importantes [4].

I.2 Analyse visuelle des données

I.2.1 Définition

La visualisation de l'information relève à la fois de la visualisation scientifique[1], du *Datamining*[2], de l'interface homme machine, de l'imagerie et des graphiques.

Il s'agit de représenter dans un espace physique sous la forme de graphiques une information souvent abstraite. Cette information peut comprendre des données, des processus, des relations ou des concepts. Sa représentation nécessite de manipuler des entités graphiques (points, lignes, formes, images, texte, surface) et leurs attributs (couleur, intensité, taille, position, forme, mouvement).

Au départ des données brutes (pas encore manipulées) sont collectées généralement grâce à l'aide d'un procédé automatisé. L'utilisateur extrait un sous-ensemble de données intéressantes organisées d'une manière plus structurée. Cette forme plus structurée peut alors être associée à une représentation visuelle par association des propriétés des données aux attributs visuels. Finalement, la représentation visuelle peut être manipulée de manière interactive par l'utilisateur en obtenant différentes vues de la même information.

Ce que Ben Shneiderman [5] nomme l'"Information seeking Mantra" - "Overview first, zoom and filter, and then details-on-demand" - est une exploration visuelle de données obéissant à un processus en trois phases :

- vue d'ensemble,
- zoom et filtrage,
- détails à la demande.

D'abord, l'utilisateur a besoin de se faire une idée de l'ensemble de données par vue d'ensemble. Il identifie par la suite des structures intéressantes et il se focalise sur une ou plusieurs d'entre elles. Enfin, pour analyser ces structures, l'utilisateur cherche à accéder au détail des données [6].

Plus de la grande quantité de données auquel on fait face dans le monde réel, ces données sont souvent multidimensionnelles. La visualisation multidimensionnelle est un sous-domaine

[1] Utilisation d'images afin de comprendre les données d'origine de mesures ou de simulation.
[2] Gestion et exploitation des données.

de la visualisation de données qui met l'accent sur plusieurs dimensions d'ensembles de données [7].

Sachant que l'humain est doté d'une capacité à visualiser l'information très développée qui joue un rôle majeur dans ses processus cognitifs (reconnaissance rapide de motifs, couleurs, formes et textures)[3] [6]. L'analyse visuelle exploite cette grande capacité humaine ce qui fait d'elle un moyen efficace pour comprendre l'information.

- La visualisation s'appuie sur les capacités incroyables et la bande passante du système visuel pour déplacer une énorme quantité d'informations dans le cerveau très rapidement,
- Elle profite de la capacité de notre cerveau à identifier les formes et communiquer les relations et les sens,
- Elle peut inspirer de nouvelles questions et une exploration plus approfondie,
- Elle permet l'identification de sous-problèmes,
- La visualisation est vraiment bonne pour identifier les tendances et les valeurs aberrantes, découvrir ou rechercher des points de données intéressantes ou spécifiques à un champ plus large.

I.2.2 Cognition [8]

Une explication sur l'efficacité de la représentation graphique a été trouvée par la psychologue américaine Anne Treisman en 1985 : la perception pré-attentive.

Percevoir le rond rouge parmi un grand nombre de ronds bleus dans la figure I-2 est immédiat et ne requiert aucun effort cognitif, quel que soit le nombre.

Anne Treisman a étudié notre perception visuelle et a découvert que nous pouvions, en regardant un graphique pendant une fraction de seconde, répondre à des questions sur le contenu de ces graphiques de manière très fiable. Par exemple, si on regarde la figure I-2 pendant 250 ms (1/4 de seconde), n'importe quelle personne ne souffrant pas d'un handicap visuel pourra percevoir qu'il y a bien un rond rouge parmi tous les ronds bleus. Si on avait mis un rond vert, on l'aurait trouvé aussi rapidement. Anne Treisman a montré que le système perceptif humain est capable de faire des traitements « en un clin d'œil », sans effort et cela de manière indépendante du nombre d'objets affichés. Lorsque des données sont affichées de manière adéquate, l'œil humain peut percevoir un grand nombre de propriétés

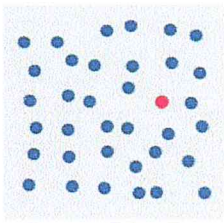


Figure I-2 : Test de cognition [8]

[3] Processus de compréhension et de mémorisation

sans effort, quelle que soit la quantité de ces données. Si on n'utilise pas une représentation adéquate, il faudra alors un temps proportionnel au nombre d'objets à étudier, ce qui devient vite pénible voire impossible. Nous pouvons percevoir un grand nombre de caractéristiques visuelles de manière pré-attentive : la couleur, l'orientation, les lignes de front et bien d'autres.

Encore une fois, les psychologues nous donnent de très bonnes raisons : les limites de notre mémoire à court terme. En effet, nous disposons de plusieurs types de mémoire. Lorsque nous menons une réflexion, nous nous reposons sur notre mémoire à court terme, ou mémoire de travail, qui est très limitée. D'après une étude de George A. Miller publiée en 1956, on ne peut mémoriser que sept items, plus ou moins deux selon les personnes et les états. Le problème est que l'expression d'une requête dans un langage comme SQL va utiliser plusieurs items de notre mémoire de travail, car notre capacité langagière alors mise à contribution a besoin de stockage. En revanche, le fait de pointer et de déplacer le curseur n'utilise aucun item. Par conséquent, nous pouvons échafauder des plans bien plus complexes et explorer plus d'alternatives avec des requêtes dynamiques qu'avec des langages de requêtes. Nous pouvons alors explorer un grand nombre de visualisations dont les paramètres sont modifiés interactivement et dont la lecture est rapide car pré-attentive. Nous avons établi une boucle de rétroaction qui nous permet d'explorer rapidement notre espace de données.
Au passage, nous pouvons voir facilement apparaitre des informations que l'on ne cherchait pas, mais que l'on trouve de manière fortuite et qui s'avère utile.

I.2.3 Objectif de la visualisation

Selon Friedman (2008) «l'objectif principal de la visualisation de données est de communiquer clairement et efficacement des informations par des moyens graphiques» [9].

Il s'agit de fournir à l'utilisateur une compréhension qualitative du contenu de l'information.

- Cette visualisation doit permettre à l'utilisateur final de faire des découvertes, proposer des explications ou prendre des décisions.
- Ces actions peuvent se faire aussi bien sur des motifs (clusters, tendances, émergences, anomalies) ou sur des ensembles d'éléments ou encore sur des éléments isolés.
- Gilles Balmisse (2005) dit encore que recourir aux technologies de la visualisation a un double objectif :
- Communiquer efficacement des informations à travers une représentation graphique via des cartes cognitives.

- Faciliter la découverte de connaissances grâce à une représentation graphique issue de l'analyse d'un corpus d'informations via des cartes sémantiques [6].

I.2.4 L'interaction humaine [6]

La visualisation de l'information ne peut être traitée sans aborder l'interaction. Cette dernière rend possible l'exploitation réelle des vues d'ensemble une fois produites. En effet, la perception est indissociable de l'action : c'est le couplage « action perception ». Ainsi l'être humain est plus habile à extraire des informations d'une interface s'il peut agir directement et activement sur cette interface que s'il reste passif.
L'interaction sera donc mise en avant dans les diverses approches de visualisation développées plus bas.

Ces dernières sont développées par Daniel Keim (2002) qui distingue les qualificatifs "dynamique" et "interactif" selon que les modifications apportées à la visualisation des données soient effectuées automatiquement ou manuellement (l'utilisateur final pouvant agir directement) :

- Projections dynamiques :

Il s'agit de changer dynamiquement les projections afin d'explorer un ensemble de données multidimensionnelles.

- Filtrage interactif :

Il s'agit d'avoir, d'une part, la possibilité de diviser interactivement l'ensemble des données dans des segments et, d'autre part, de se concentrer sur les sous-ensembles intéressants. Ceci peut être fait en choisissant directement le sous-ensemble désiré (browsing) ou en spécifiant des propriétés du sous-ensemble désiré (querying).

- Zoom interactif :

Il s'agit de partir d'une vue globale des données et de permettre l'affichage des détails selon différentes résolutions.

- Distorsion interactive :

Il a l'avantage de pouvoir montrer des parties de données avec un niveau élevé de détail tandis que la vue d'ensemble est préservée (les autres parties de données étant visibles avec un niveau moins détaillé). Il existe des techniques de déformation hyperboliques ou sphériques

souvent employées sur des hiérarchies ou des graphiques. Ces techniques utilisent une sorte de loupe (fisheye) déformante que l'on promène à son gré sur l'ensemble des données.

- Liens interactifs et brossage (interactive linking and brushing)

Pour les données multidimensionnelles : l'idée est de combiner des méthodes différentes de visualisation pour surmonter les imperfections des techniques simples.
Ces techniques d'interaction permettent de définir les tâches de l'utilisateur. Selon Ben Shneiderman, les tâches interactives possibles par l'analyste sont :

- avoir une vue de l'ensemble ou globale
- zoomer
- filtrer
- détailler
- voir les relations entre objets
- avoir l'historique des actions pour le rejouer
- extraire

I.2.5 Type de visualisation

D'après le dictionnaire CNRTL (Centre National de Ressources Textuelles et Lexicales), la visualisation est la présentation visuelle sur un écran des résultats d'un traitement sous forme alphanumérique ou graphique. Card et al (1999) quant à eux définissent le terme de visualisation par l'utilisation, assistée par l'ordinateur, des représentations visuelles de données pour amplifier la cognition. Ces définitions montrent que la visualisation est une activité cognitive facilitée par une représentation graphique externe pour aider les utilisateurs et les analystes de construire une représentation mentale interne sur le monde [10]. Il existe trois catégories de la visualisation qui sont les suivantes :

- La visualisation scientifique : permet de comprendre les phénomènes physiques dans les données et qui se base sur des modèles mathématiques,
- La visualisation d'information : vise à explorer les données et les informations sous forme graphique qui permet aussi d'extraire et d'identifier les tendances, les corrélations et les structures abstraites dans les données [11],
- La visualisation de connaissances : est utilisée pour désigner tout procédé permettant de présenter une structure de connaissance comme moyen pour évaluer soi-même des connaissances et aider à la compréhension et à la navigation [1].

Dans notre cadre d'étude, on se concentre sur la visualisation d'information que nous allons développer dans ce qui suit.

I.2.6 Domaine d'application

L'analyse visuelle est essentielle dans des domaines où les grands espaces de l'information doivent être traités et analysés. Domaines d'application principaux sont la physique et l'astronomie.

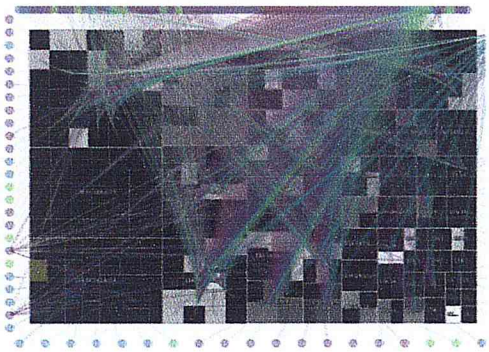


Figure I-3 : Analyse d'une attaque de réseau distribué sur le service SSH d'un réseau universitaire [4]

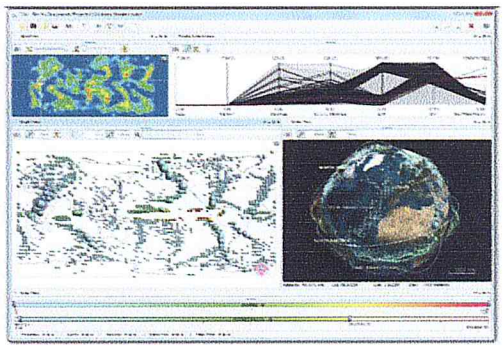


Figure I-4 : Support visuel pour la simulation de modèles climatiques [4]

Par exemple, la discipline de l'astrophysique offre de nombreuses possibilités pour les techniques d'analyse visuelle : volumes massifs de données non structurées, provenant de différentes directions de l'espace et couvrant l'ensemble du spectre de fréquence, de flux continus de téraoctets de données qui peuvent être enregistrées et analysées. Avec les techniques d'analyse de données commune , les astronomes peuvent séparer les données pertinentes du bruit , analyser les similitudes ou des motifs complexes , et d'acquérir des connaissances utiles sur l'univers, mais l'approche de l'analyse visuelle peuvent soutenir de manière significative le processus d'identification des phénomènes inattendus à l'intérieur du massif des données et dynamique des cours d'eau qui autrement ne serait pas trouvée par des moyens algorithmiques standards . Climat et le temps de surveillance est également un domaine qui implique d'énormes quantités de données recueillies par les capteurs à travers le monde et à partir des satellites, de courts intervalles de temps. Une approche visuelle peut aider à interpréter ces énormes quantités de données et de mieux comprendre les facteurs climatiques dépendantes et les scénarios de changement climatique qui seraient autrement pas être facilement identifiés. Outre les prévisions météorologiques, les applications existantes visualiser le réchauffement climatique, la fonte des pôles, l'appauvrissement de l'ozone stratosphérique, ainsi que les ouragans et tsunamis avertissements.

Dans le domaine de la gestion des urgences, l'analyse visuelle peut aider à déterminer les progrès en cours d'une urgence et d'identifier les prochaines contre-mesures (par exemple, la construction de contre-mesures physiques ou évacuation de la population) qui doivent être prises pour limiter les dégâts. Ces scénarios peuvent inclure les catastrophes naturelles ou météorologiques comme les inondations ou les vagues, les volcans, la tempête, le feu ou la croissance épidémique de maladies (virus N1H1 par exemple), mais aussi les catastrophes technologiques par l'homme comme les accidents industriels, les accidents de transport ou de pollution.

L'analyse visuelle pour la sécurité et la géo- graphiques est un sujet de recherche important. Le champ d'application dans ce secteur est large, allant de l'informatique de terrorisme, la protection des frontières, la détection de chemin à la sécurité du réseau. L'analyse visuelle soutient la détection des similitudes et des anomalies dans de très grands ensembles de données. Par exemple, à l'échelle mondiale, par jour, il y a plus de 210 milliards de courriels, 4 milliards de SMS, 90 millions de tweets et le nombre de paquets de données IP dépasse 9000 milliards.

En biologie, en médecine, la tomographie par ordinateur et l'imagerie par ultrasons pour la reconstruction numérique en 3 dimensions et de visualisation produisent des giga-octets de données médicales. Le domaine d'application de la bio-informatique utilise des techniques d'analyse visuelle pour analyser de grandes quantités de données biologiques. Dès le début de séquençage, les scientifiques dans ces domaines font face à des volumes de données sans précédent, comme dans le projet du génome humain avec trois milliards de paires de base par l'homme. Autres domaines nouveaux comme la protéomique[4], la métabolomique[5] ou la chimie combinatoire avec des dizaines de millions de composés, ajouter des quantités importantes de données chaque jour. Un calcul de toutes les combinaisons possibles de force brute n'est souvent pas possible, mais les approches visuelles interactives peuvent aider à identifier les principales régions d'intérêt et exclure les zones peu prometteuses.

Un autre domaine majeur d'application pour l'analyse visuelle est le business intelligence. Le marché financier avec ses centaines de milliers de biens génère de grandes quantités de données sur une base quotidienne, ce qui conduit à des volumes de données très élevés au cours des années. Par exemple, on estime qu'il y a plus de 300 millions de transactions Visa par carte de crédit par jour. Le principal défi dans ce domaine est d'analyser les données

[4] Étude des protéines dans une cellule.
[5] Étude systématique des empreintes digitales chimiques uniques que les processus cellulaires spécifiques laissent derrière eux.

- Données multidimensionnelles

De nombreux ensembles de données se compose de plus de trois attributs et par conséquent, ils ne permettent pas une simple visualisation sous forme de diagrammes 2 ou 3 dimensions. Un exemple de données multidimensionnelles (ou multi-variées) sont des tables de bases de données relationnelles, qui ont souvent des dizaines voire des centaines de colonnes (ou attributs). Comme il n'existe pas un simple mappage des attributs sur les écrans de nature 2 dimensions, des techniques de visualisation plus sophistiquées sont nécessaires. Une des techniques qui permet la visualisation de données multidimensionnelles est les Coordonnées parallèles, qui est également utilisé dans le cadre évolutive.

- Texte et hypertexte

La description des données n'est pas limitée aux dimensions qui la composent. À l'ère de l'internet, on reconnait un type de données important qui est le texte et l'hypertexte, ainsi que le contenu multimédia des pages web. La différence de ces données est qu'elles ne peuvent pas être décrites par des nombres et par conséquent, la plupart des techniques de visualisation standard ne peuvent pas être appliquées. Dans la plupart des cas, une première transformation de ces données en des vecteurs est nécessaire avant l'utilisation des techniques de visualisation. Un exemple des techniques de transformation est le comptage des mots qui est souvent combiné avec une analyse multidimensionnelle.

- Hiérarchies et graphiques

Les enregistrements ont souvent un rapport avec d'autres éléments d'information. Les graphiques sont largement utilisés pour représenter ces interdépendances. Un graphique est composé d'un ensemble d'objets, appelés nœuds, et des liens entre ces objets, appelés arêtes. Les interactions e-mail entre les gens, leur comportement d'achat, la structure des fichiers du disque dur ou les liens hypertexte sur internet font partie de ce type de données. Il y a un certain nombre de techniques de visualisation spécifiques qui traitent des données hiérarchiques et graphiques. Un des outils de visualisation de ce type de données est le framework Scalable.

- Algorithmes et logiciels

Une autre classe de données est les algorithmes et les logiciels. Faire face à de grands projets de logiciels est un défi. Le but de la visualisation est de soutenir le développement de logiciels en les aidant à comprendre les algorithmes, par exemple, en montrant la circulation de l'information dans un programme et en amélioreant la compréhension du code écrit, par

exemple, en représentant la structure de milliers de lignes du code source sous forme de graphiques, et de soutenir le programmeur dans le débogage du code, c'est à dire par des erreurs de visualisation. Il existe un grand nombre d'outils et de systèmes qui soutiennent ces tâches comme Polaris.

I.2.9 Technique de visualisation

Il existe un grand nombre de techniques de visualisation des données. En plus de 2D/3D-techniques standard telles que xy et xyz parcelles, histogrammes, des graphiques linéaires, etc., il y a un certain nombre de techniques de visualisation plus sophistiquées. Les classes correspondent aux principes de visualisation de base qui peuvent être combinées afin de mettre en œuvre un système de visualisation spécifique :

- Affichage géométriquement transformées,
- Affichage emblématiques,
- Orienté pixel,
- Affichage empilés,
- Projection dynamique [1].

Dans le processus de visualisation, il convient de tenir compte du choix de la meilleure technique qui sera utilisée dans certaines applications ou situations. L'utilisation inadéquate des techniques de visualisation peut générer des résultats insuffisants ou même incorrectes, causés par des erreurs de représentation graphique.
Il existe de nombreuses visualisations et un nombre important de taxonomies [5]. Nous présentons quelques-unes de ces techniques :

- 2D et 3D Scatterplots (Nuage de points)

Un nuage en projection du point (généralement affine) des données dans un espace de dimension 2D ou 3D représenté sur l'écran dans le classique (X, Y) ou (X, Y, Z) le format. C'est la méthode la plus couramment utilisée de la visualisation des données les figures I-7, I-8 montre la représentation de Scatterplots.

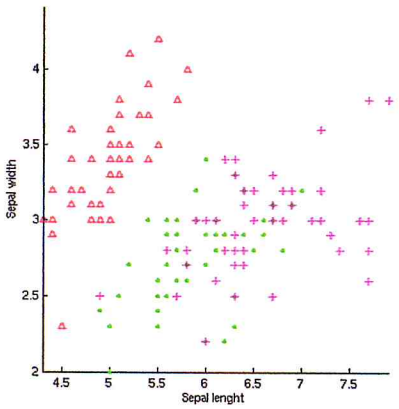


Figure I-7 : Nuage de points en 2D [14]

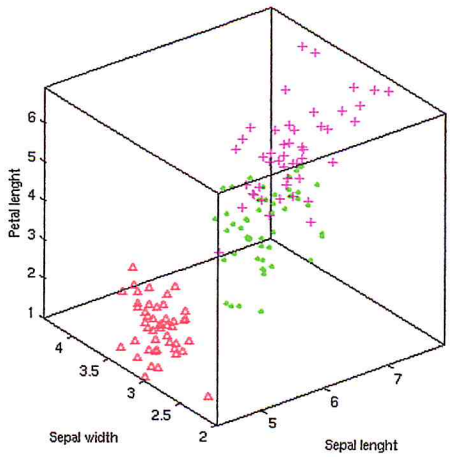


Figure I-8 : Nuage de points en 3D [14]

De nombreuses applications ou des transformations peuvent être appliquées. Les points affichés peuvent avoir de nombreux attributs tels que la couleur, la taille, la forme, la texture, le mouvement et même le son (quand on interagi avec). Pour interpréter l'interaction de projection 3D, il est nécessaire de résoudre les ambiguïtés, bien que d'autres techniques ont été utilisées (animation) [14].

- La matrice de Scatterplots :

Une matrice de nuages de points (figure I-9) est un tableau de diagrammes de dispersion présentant toutes les paires possibles de dimensions ou coordonnées. Pour les données de dimension n on obtient $n(n-1)/2$ nuages de points avec des échelles communes, bien que le plus souvent n^2 nuages de points sont affichés. Les nuages de points peuvent également être placés dans un autre format que les tableaux (circulaire, hexagonale, etc.) On peut relier visuellement les caractéristiques d'un nuage de points avec des fonctionnalités sur un autre, ce qui augmente considérablement sa puissance [14].

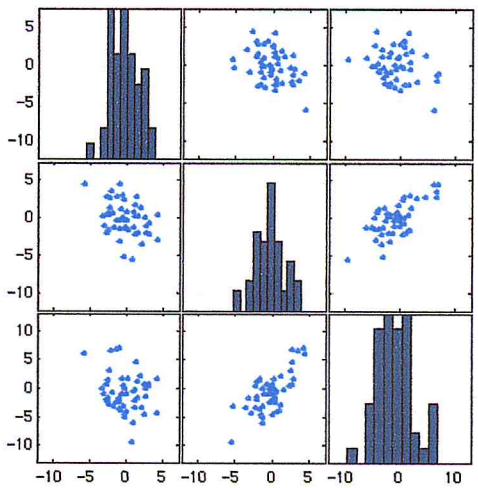


Figure I-9 : Matrice de nuage de points [14]

➢ Heat maps (cartes de chaleurs)

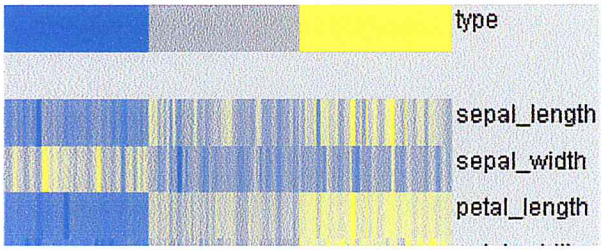


Figure I-10 : Carte de chaleurs [14]

Il s'agit d'un réseau de cellules où chaque cellule est colorée en fonction de certaines valeurs de données ou d'une fonction sur les données. La méthode est une généralisation de Scatterplots où les points sont des cellules de la grille, et chaque cellule est colorée. Il existe de nombreuses variantes nommées (de la carte cluster d'image, Heatmaps, patchgrid) [14].

➢ Height maps (Carte de hauteurs)

Les carte de hauteur (figure I-11) est une nouvelle extension de la carte de chaleur avec la grille représentée comme un champ de hauteur au lieu de par la couleur. Faire de la taille de petites cellules peut générer une carte presque continue [14].

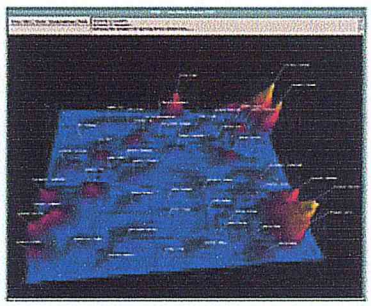


Figure I-11 : Carte de hauteurs [14]

➢ Parallel Coordinates (Les coordonnées parallèles)

Les coordonnées parallèles (figure I-12) se servent d'axes parallèles perpendiculaires pour la représentation d'ensemble de données multidimensionnel [15].

Dans cette approche, chaque dimension est dessinée comme une ligne verticale, et chaque point multidimensionnel est visualisé comme une poly-ligne qui traverse chaque axe à la position appropriée pour refléter la valeur d'une donnée sur chaque dimension [16].

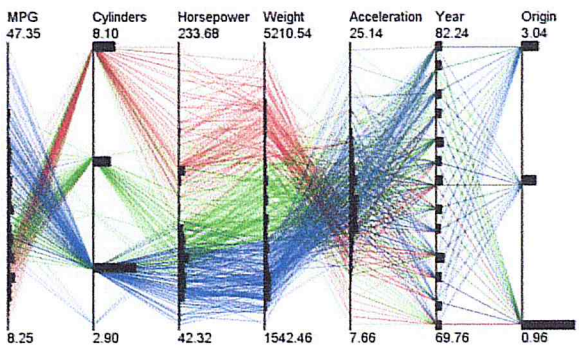


Figure I-12 : Les coordonnées parallèles [16]

Les valeurs maximales et minimales de chaque dimension généralement mises à l'échelle à des limites supérieures et inférieures sur ces lignes verticales.

➢ Polar chart (Diagramme polaire)

Un diagramme polaire (Figure I-13) est un graphique circulaire pour le traçage des coordonnées polaires. Les Coordonnées polaires pour mappe les données sur une surface 2D en

utilisant l'angle et le rayon, la création d'une version "wrap-around" d'un graphique linéaire. Polar chart dépasse la limitation des graphiques linéaires, qui sont utilisés uniquement pour l'affichage d'une valeur unique ou par morceaux de fonctions continues d'une dimension. Ceux-ci peuvent être considérés comme des représentations circulaires de coordonnées parallèles et peuvent donc réduire l'effet de limitation d'un grand nombre de dimensions. Toutefois, la taille des représentations de points de données dépend de la proximité du centre [14].

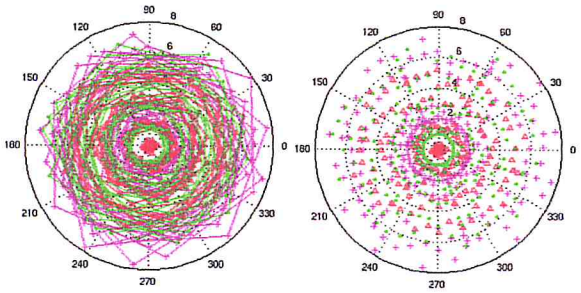


Figure I-13 : Diagramme polaire [14]

- RadViz

RadViz (figure I-14) est une technique d'affichage qui met les ancres de dimensions autour du périmètre d'un cercle. Des étiquetés sont utilisés pour représenter des valeurs relationnelles entre les points - une extrémité d'une étiquetés est fixé à une ancre dimensionnelle, l'autre est attaché à un point de données. Les valeurs de chaque dimension sont généralement normalisées à 0 à1 gamme. Chaque point de données est affichée à l'endroit où la somme de toutes les forces de ressort est égale à zéro. La position d'un point de données dépend en grande partie de l'agencement de dimensions autour du cercle [14].

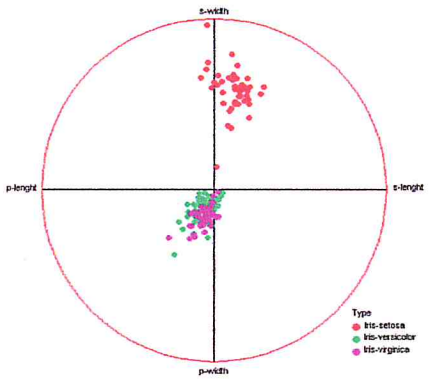


Figure I-14 : RadViz [14]

I.2.10 Outils de visualisation

- GGobi

Ggobi est un programme open source de visualisation pour l'exploration des données de grande dimension. Il fournit des graphes dynamiques et interactifs tels que des visites guidées, ainsi que des graphes familiers tels que *scatterplot*, *barchart* et *parallel coordinates plots* [17].

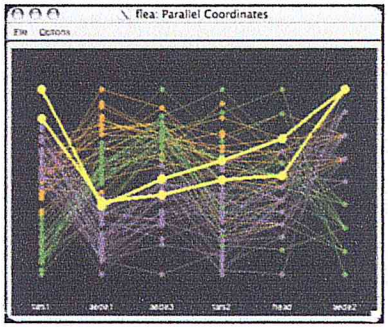


Figure I-15 : Les coordonnées parallèles sur GGobi [17]

- ROOT

Est un programme et bibliothèque orienté objet développé par le *CERN* (Organisation européenne pour la recherche nucléaire). Il a été initialement conçu pour l'analyse de données de physique des particules et contient plusieurs fonctionnalités spécifiques à ce domaine, mais il est également utilisé dans d'autres applications telles que l'astronomie et l'exploration de données [18].

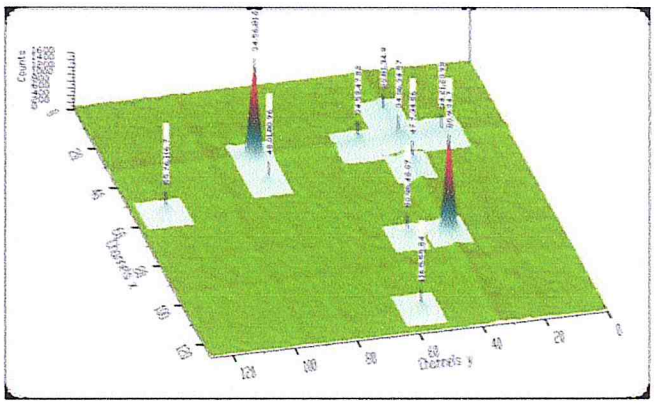


Figure I-16 : Les cartes de hauteur sur ROOT [18]

- Macrofocus High-D

Macrofocus Haute-D est un logiciel de visualisation d'informations sur la base de coordonnées parallèles utilisant les données de base de logiciels commun (par exemple, Excel, Access), les bases de données SQL ou les sources de données en ligne comme Yahoo! Finance. Principales caractéristiques comprennent la capacité facile d'organiser l'axe, pour filtrer par la recherche ou de la valeur, et de recevoir des informations détaillées en survolant dans des parties de l'intrigue parallèle coordonnées.

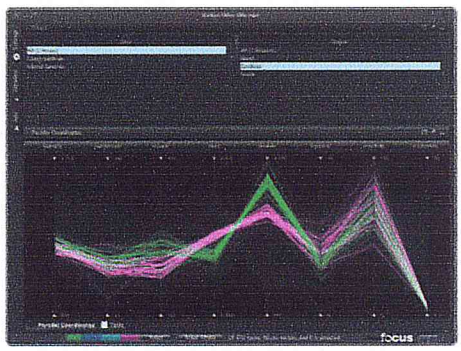


Figure I-17 : Les coordonnées parallèles sur Marcofocus [19]

Macrofocus Haute-D est basé sur Java et donc compatible avec Microsoft Windows, Mac OS X, Linux [19].

- Spotfire

Développé par la société TIBCO, ce logiciel donne la possibilité aux utilisateurs d'analyser leurs données à l'aide des graphes. Il est spécialisé dans le domaine de business intelligence.

La dernière version développée est TIBCO Spotfire 6.5 [20].

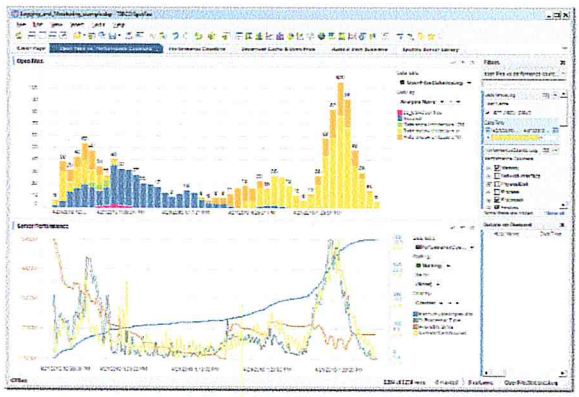


Figure I-18: Interface de Spotfire [20]

- Orange

Un logiciel open source de visualisation de données et d'analyse pour les débutants et les experts. L'exploration de données grâce à une programmation visuelle ou des scripts Python. Il possède des composants pour l'apprentissage de la machine et des modules pour la bio-informatique et de texte-mining. Doté de fonctions pour l'analyse de données [21].

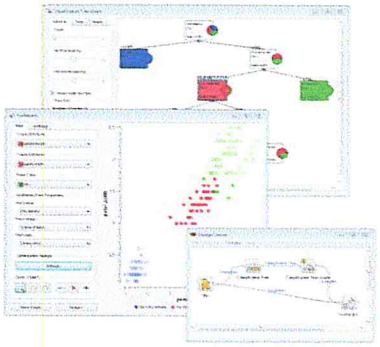


Figure I-19: La représentation des données sur Orange [21]

- Xmdv

XmdvTool est un logiciel d'application public pour l'exploration visuelle interactive d'ensembles de données multidimensionnelle. Il est disponible sur toutes les plateformes majeures telles qu'UNIX, LINUX, MAC et Windows. XmdvTool est développé en utilisant Qt et Eclipse CDT. Il prend en charge cinq méthodes pour afficher des données de forme plate et données hiérarchiquement cluster (Scatterplots, Star Glyphs, Parallel Coordinates, Dimensional Stacking, Pixel-oriented Display) [22].

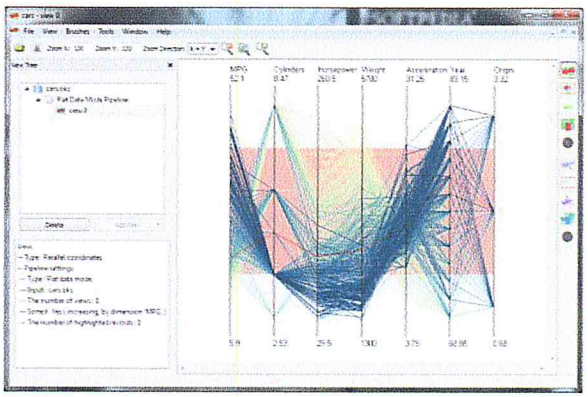


Figure I-20 : Les coordonnées parallèles sur Xmdv [22]

Le besoin le plus important dans notre travail est que la technique de visualisation soit multidimensionnelles applicable sur n'importe quel type de données (continue et catégoriel). Tout au long de notre projet, nous travaillerons sur les coordonnées parallèles qui est l'une des techniques les plus performantes et plus simple en même temps. Sa façon de représenter les

données et les dimensions d'analyse offre la possibilité d'analyser un grand nombre de données. Quant à sa structure visuelle, elle permet de visualiser la relation qui existe entre ces données. Ce qui induit à l'extraction de connaissances [23].

I.3 Les coordonnées parallèles

I.3.1 Définition

La technique des coordonnées parallèles (ParCoords) a été redécouverte[6] par inselberg en 1985 en tant que mécanisme pour l'étude géométrique de grande dimension. Depuis lors, de nombreuses recherches, y compris inselberg, ont étudié et amélioré ParCoords pour l'analyse de données multidimensionnelles. L'idée de base est que les axes, au lieu d'être orthogonale, sont parallèles, avec des lignes verticales ou horizontales régulièrement espacées dans un ordre particulier des dimensions [24].

Le ParCoords, qui transforme les données multidimensionnelles en poly-lignes 2D, a été largement utilisé dans de nombreuses applications de visualisation d'informations, ainsi que l'exploration de données [23].

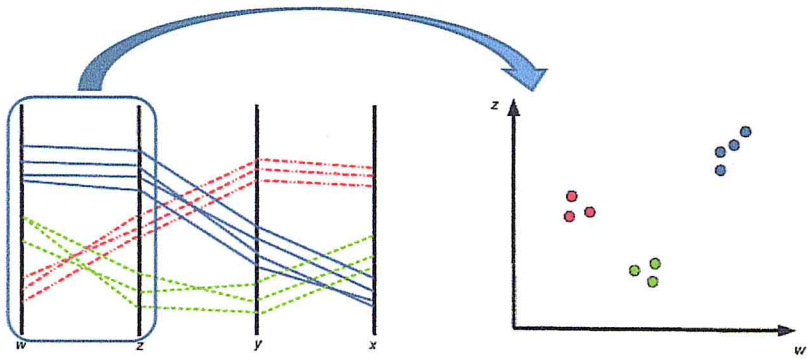


Figure I-21 : La projection de ParCoors sur un graphe classique à 2D [23]

La figure I-20 montre qu'une ligne entre deux axes sur les coordonnées parallèles est dessinée comme un point dans le graphe classique a deux dimension (w,z).

I.3.2 Visualisation des clusters par les coordonnées parallèles

Dans notre travail, nous utilisons pour la représentation des données les coordonnées parallèles qui permettent de visualiser simultanément un nombre de dimensions important.

Pour interpréter le graphe, on cherche des clusters de lignes similaires (indiquant corrélation partielle entre des paires de dimensions), les points de croisement similaires (indiquant une

[6] Coordonnées parallèles a été inventer par Philbert Maurice d'Ocagne en 1885 [72].

corrélation négative), et des lignes isolées ayant une valeur sensiblement différente de leurs voisins (indiquant valeurs aberrantes). Un des points forts de ParCoords est sa capacité à montrer la corrélation entre les dimensions.

De nombreux chercheurs ont travaillé sur le développement des capacités de ParCoords. Parmi les techniques proposées :

- Le ParCoords hiérarchiques qui montre les clusters au lieu des données,
- Utilisation des lignes semi-transparentes pour révéler les clusters dans de grands ensembles de données,
- Le *Clustering*, l'ordonnancement, et l'espacement des axes basé sur la corrélation,
- Ordonnancement des axes pour réduire l'encombrement d'affichage,
- Regroupement des données dans des clusters avec traitement des valeurs aberrantes,
- Incorporation des histogrammes dans les axes pour mieux communiquer les distributions uni-variées,
- Ajustements des courbes aux points d'intersection pour exprimer la continuité entre axes [24].

D. Keim a proposé une application sur des données boursières, ce qui permet de voir les tendances ou les anomalies.

Le but de graphe généré par l'application est essentiellement exploratoire : pour faire découvrir un phénomène qui est difficilement détectable [6].

Wegman [25] montre que les coordonnées parallèles peuvent être utilisées pour révéler efficacement la corrélation de données ainsi que l'interaction des clusters. Cette observation conduit à la découverte de connaissances importante [23].

I.3.3 Techniques de représentation des clusters

Wegman [25] a développé les coordonnées parallèles dans les aspects de la géométrie, statistiques et graphiques, qui ont été largement appliquées dans la visualisation d'information. Pour visualiser des ensembles de données en cluster, de nombreuses approches ont été menées en utilisant des coordonnées parallèles [23].

- Vision hiérarchique

Chaque nœud T_i dans un arbre de classification hiérarchique T représente un ensemble imbriqué de points de données ou sous-groupes. A chaque nœud, en maintenant des renseignements

sommaires de tous les points et sous-clusters enracinés en lui. Les informations suivantes peuvent être obtenues directement à partir deT_i.

- n_i : le nombre de points de données joint.
- m_i : la moyenne des points de données.
- B_i : l'étendue, c'est à dire le minimum et le maximum des bornes de cluster pour chaque dimension.
- v_i : une mesure de la taille de cluster T_i.
- l_i : la profondeur de l'arbre au nœud T_i.

v_i est une mesure calculée de la taille de cluster et satisfait le critère suivant :

Si T_i est un ancêtre de T_j, alors $v_i > v_j$

La valeur de v_i dépend directement de la forme des clusters produits par l'algorithme de *Clustering*. Pour les clusters sphériques, v_i peut-être le rayon d'un cluster. Pour les clusters rectangulaires, v_i peut-être le volume à N dimensions de la grappe.

Ying.H et al [26] proposent de représenter l'information à un nœud en faisant usage de bandes d'opacité de largeur variable. La figure I-21 montre une bande graduée fané d'un milieu dense à bords transparents qui codent visuellement les informations d'un cluster.

Le moyen s'étend à travers le milieu de la bande et est codée avec l'opacité la plus foncée, qui est une fonction de la densité d'un cluster, définie comme le rapport $\frac{n_i}{v_i}$.

Cela permet de faire la différence de répartition, denses, étroites ou grand clusters. Les bords supérieurs et inférieurs de la bande ont une transparence totale. L'opacité dans le reste de la bande est interpolée linéairement. L'épaisseur de la bande à travers chaque section de l'axe représente l'étendue du cluster dans cette dimension [26].

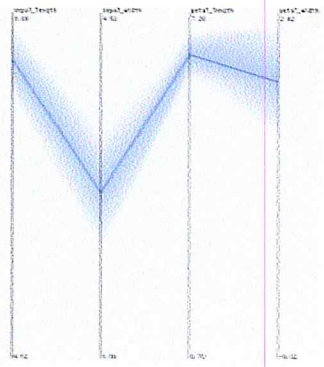


Figure I-22 : Vision hiérarchique d'un cluster sur ParCoors [26]

➢ Modèle de réduction d'énergie

Yang.X et al [23] innovent un modèle de réduction d'énergie quadratique pour construire la forme de cluster. Ce modelé permet de réduire l'encombrement visuel tout en préservant les détails essentiels de chaque cluster, en associant à chaque ligne i (avec z_i étant son centre) entre deux dimensions x et y avec un effet de «bande de caoutchouc» avec trois adjacent énergie potentielle: énergie élastique, énergie d'attraction, énergie répulsif.

Énergie élastique : $E_E(i) = (z_i - \frac{x_i+y_i}{2})^2$ ..... (1)

Énergie d'attraction : $E_A(i, \hat{c}_p) = (z_i - \hat{c}_p)^2$ .....(2)

Énergie répulsif : $E_R(i, \hat{c}_{p-1}, \hat{c}_{p+1}) = (z_i - \hat{c}_{p-1})^2 + (z_i - \hat{c}_{p+1})^2$ ...(3)

Ici, chaque groupe dispose d'un centre d'attraction qui peut servir de centre de répulsion pour ses clusters voisins.

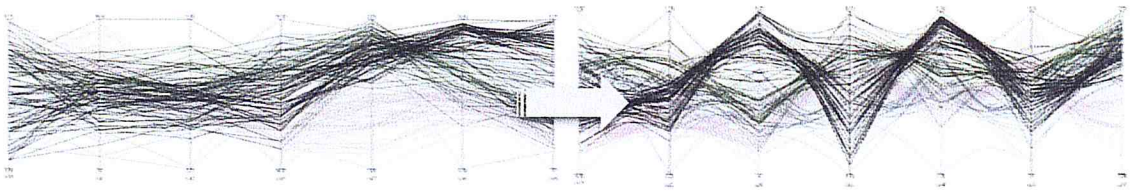


Figure I-23 : l'effet de réduction d'énergie sur les clusters [27]

Une autre technique de réduction d'énergie a été proposée par Hong.Z et al [27]. Dans ce modèle, l'énergie totale des arêtes peut être divisée en deux termes majeurs tels que représenté par l'équation suivante:

$$E = \alpha_c Ecurvature + (1 - \alpha_c) Egravitation \ldots. (1)$$

Où E est l'énergie totale de l'ensemble du système, Ecurvature est le terme d'énergie représentant la flexion de chaque ligne, Egravitation est le terme d'énergie représentant les interactions entre les paires de lignes voisines.

- Coloration

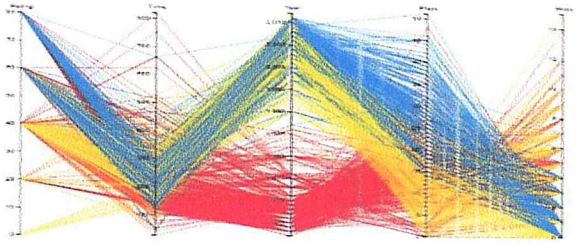


Figure I-24: Coloration des clusters sur les coordonnées parallèles [28]

Une méthode simple qui attributs une couleur à chaque cluster, ainsi les poly-lignes qui illustre les données de même cluster aurons une couleur qui les caractérise.

- Min-Max

Pour représenter les clusters, cette méthode utilise la valeur minimum et maximum des données qui regroupent sur chaque dimension pour désigner les polygones représentatifs.

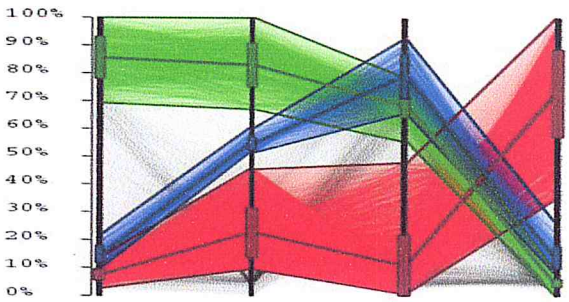


Figure I-25 : La representation des clusters par MinMax [71]

I.4 Problématique

Le ParCoords souffre de plusieurs problèmes, en particulier lors de la représentation des grands ensembles de données. Une représentation potentiellement lourde à interpreter, ce qui entraîne un encombrement visuel [29]. La visualisation des clusters qui regroupe les données est une des methodes d'améliorations de visualisation qui réduit l'encombrement d'affichage. De même, l'ordre des dimensions en ParCoords peut avoir un impact majeur sur l'expressivité de la visualisation et affecter l'encombrement perçu des données et la relation entre elles, ce qui provoque des conclusions complètement différentes établies sur la base de chaque représentation. Malheureusement, dans de nombreux systèmes de visualisation existants qui englobent ces techniques, les dimensions sont généralement ordonnées sans beaucoup de traitement. En fait, les dimensions sont souvent affichées dans un ordre par défaut [7].

La figure I-25 indique que différentes permutations des n dimensions montrent différents aspects de l'ensemble de données. Cet ensemble de données avec des clusters définit, différentes

permutations donnent différents points de vue sur les relations de ces groupes et peut changer l'interprétation des données.

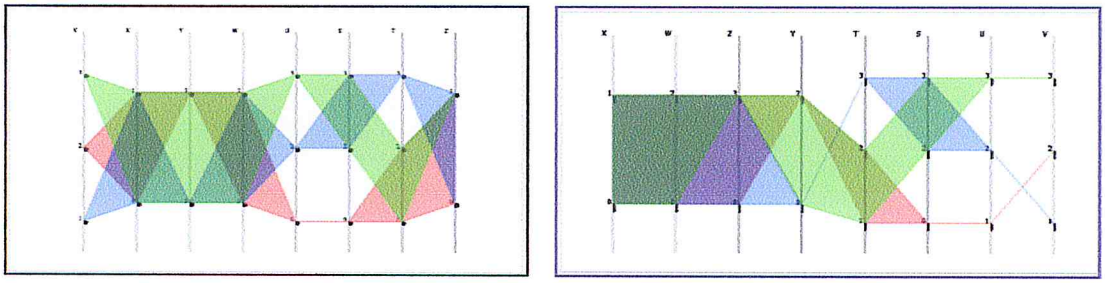


Figure I-26 : L'effet d'ordonnancement des dimensions sur la visualisation des clusters [30]

La figure I-26 indique différente représentation des mêmes données avec changement d'ordonnancement des dimensions. Ça illustre l'effet de l'ordre des axes sur la représentation.

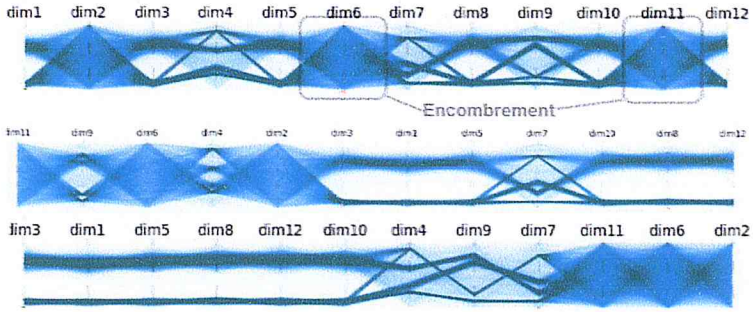


Figure I-27 : L'effet d'ordonnancement des dimensions sur la visualisation des données [31]

Un ordonnancement manuel est disponible dans certains systèmes. Par exemple, Polaris permet aux utilisateurs de sélectionner et de commander les dimensions pour être mappé à l'écran manuellement. De même, dans XmdvTool, les utilisateurs peuvent modifier manuellement l'ordre des dimensions à partir d'une liste reconfigurable de dimensions. Cependant, la recherche exhaustive du meilleur ordre est fastidieuse, même pour un petit nombre de dimensions. Pour vérifier tous les ordres de dimensions possibles $(n-1)!$ étant n le nombre de dimensions. En outre, l'utilisateur aura du mal à se souvenir de tous les points de vue et de l'ordre des dimensions qu'il a visionné [7]. Par conséquent, Pour faire de ParCoords une technique d'analysée visuelle efficace, il est nécessaire de présenter les dimensions dans un meilleur ordre qui aide l'utilisateur dans ses tâches d'analyse.

Il en résulte de tout ce qui a été dit que notre intérêt premier était d'offrir à l'analyste le meilleur ordre des dimensions de coordonnées parallèles pour une visualisation optimisée, et une interprétation et extraction plus facile des connaissances.

I.5 Conclusion

Ce chapitre nous a permis de comprendre l'analyse visuelle des données, ces avantages dans la représentation, la compréhension des données, et la découverte de nouvelles connaissances à partir de celle-ci. Aussi, de réaliser les déficits de ces techniques spécialement le ParCoords et les possibilités d'amélioration.

Notre travail consiste à améliorer le ParCoords en réordonnançant les dimensions d'analyse. Le problème de trouver un ordre de permutation optimale selon une mesure spécifique ressemble fortement au problème du voyageur de commerce. Chaque sommet du graphe est une dimension et chaque arête dans le graphe est une paire possible de dimensions. Chaque arête possède une valeur qui lui est associée selon une mesure spécifique, par exemple, la mesure de corrélation de Pearson entre deux dimensions. Dans un problème de voyageur de commerce régulier, les arêtes sont généralement associées à des valeurs de distance et le chemin à trouver est une boucle fermée (cycle Hamiltonien). Dans le problème d'ordonnancement de dimension, le chemin à trouver est une "route" ouverte (Chemin Hamiltonien dans un graphe) [32].

Chapitre

II. Problème d'arrangement des axes

II.1. Introduction :

Le problème du voyageur de commerce ou *Traveling Salesman Problem* (TSP) est largement étudié en informatique. Il existe de nombreuses publications à son propos, depuis 1940 jusqu'à les plus récentes. Le TSP a suscité les intérêts des informaticiens et des mathématiciens, car même après environ une demi-décennie de recherche, le problème n'a pas été complètement résolu (il n'existe pas de solution qui s'exécute en temps polynomiale). Ce problème tombe dans une catégorie unique de problèmes NP-difficiles. Il peut être appliqué à résoudre de nombreux problèmes pratiques dans notre vie quotidienne. Ainsi, une solution au TSP serait très bénéfique.

II.2. Définition :

Le TSP est définit comme suit : étant donné un graphe complet, G, avec un ensemble de sommets, V, un ensemble d'arêtes, E, et un coût, c_{ij}, associé à chaque arête dans E. La valeur c_{ij} est le coût engagé lors de la traversée du sommet $i \in V$ vers le sommet $j \in V$. Compte tenu de ces informations, une solution au TSP doit retourner le cycle Hamiltonien le moins cher de graphe G. Un cycle Hamiltonien est un cycle qui visite chaque nœud dans un graphe exactement une fois et retourne au sommet de départ. Par contre un chemin Hamiltonien est ouvert et son sommet final n'est pas relier au sommet de départ.

II.3. Histoire :

Le problème du voyageur de commerce (TSP) a été étudié au 18ème siècle par un mathématicien Irlandais nommé Sir William Hamilton Rowam et par le mathématicien britannique du nom de Thomas Penyngton Kirkman [33]. A croire que la forme générale de la TSP a été d'abord étudiée par Kalr Menger à Vienne et à Harvard. Le problème a été promu par Hassler, Whitney & Merrill à Princeto (Schrijver, 1960). Puis, dans les années 1940, le mathématicien Merrill Flood Meeks, diffuse le nom, TSP, au sein de la communauté mathématique [34].

C'est durant l'année 1948, que Flood a présenté le problème du voyageur de commerce à la RAND[7] Corporation [34], selon Flood "quand je me débattais avec le problème de connexion avec une étude de routage d'autobus scolaire dans le New Jersey" (Flood, 1956). Le TSP est vite devenu très populaire. Cette popularité a été probablement attribuable à quelques facteurs, dont l'un est le prestige de la RAND Corporation. Un autre facteur est le lien entre le problème TSP et les problèmes combinatoires dans la programmation linéaire. Enfin, son titre

[7] RAND corporation est une institution américaine à but non lucratif qui a pour objectif d'améliorer la politique et le processus décisionnel par la recherche et l'analyse.

est certainement un facteur, ce qui démontre la pertinence vers de nombreuses tâches évidentes dans la vie quotidienne des gens.

Le TSP démontre tous les aspects de l'optimisation combinatoire. Pendant les années 1950, la programmation linéaire a été de plus en plus une force vitale dans des solutions informatiques à des problèmes d'optimisation combinatoire. Cela était dû au financement accordé par l'US Air Force dans le but d'obtenir des solutions optimales aux problèmes de transport combinatoires.

Les tentatives pour résoudre le TSP ont été en vain jusqu'au milieu des années 1950 quand Dantzig, Fulkerson et Johnson ont présenté une méthode pour résoudre le TSP. Ils ont montré l'efficacité de leur méthode de résolution d'un exemple de 49 villes [35].

Cependant, il est devenu évident, dès les années 1960, que l'instance générale de la TSP ne pouvait être résolue en temps polynomial en utilisant des techniques de programmation linéaire. En fait, il a été conjecturé que le TSP, et des problèmes semblables, prenaient beaucoup de temps pour la résolution et le coût de résolution augmentait de façon exponentielle avec la taille du problème, ce qui nous mène à la notion de complexité.

II.4. La complexité :

La recherche d'une solution pour le TSP consiste à parcourir une forêt d'arbres (Les sommets étant les racines de chaque arbre). Cette forêt de recherche contient l'ensemble de toutes les solutions possibles. Une branche d'un arbre représente une éventuelle solution. La hauteur d'un tel arbre est polynomiale par contre son nombre de branches est exponentiel. Pour notre problème la hauteur de l'arbre est N, et un nœud de hauteur i possède (N - i) fils.
Donc pour N villes, il y a $(N - 1)!$ chemins possibles. Comme il n'existe aucun algorithme exact qui donne la meilleure solution au problème dans un temps polynomial le TSP est classé parmi les problèmes NP-difficile avec une complexité $\Theta(n!)$ [36, 37].

II.5. Formalisation :

Formellement, le problème du voyageur de commerce peut s'écrire comme un programme linéaire. Ci-dessous, V est l'ensemble des n sommets du graphe. x_{ij} désigne l'arc (i, j) et vaut 1 s'il fait partie de la solution, 0 sinon. c_{ij} représente le poids de l'arc (i, j).

$$\begin{cases} \text{Min } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1) \\ \sum_{i=1}^{n} x_{ij} \qquad\qquad \forall\, i \in V \quad \ldots\ldots\ldots\ldots\ldots(2) \\ \sum_{j=1}^{n} x_{ij} \qquad\qquad \forall\, j \in V \quad \ldots\ldots\ldots\ldots\ldots(3) \\ \sum_{(i,j)\in P^2} x_{ij} \leq |P| - 1 \quad \forall\, P \subset V, P \neq \emptyset \ldots(4) \\ x_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in V^2 \end{cases}$$

(2) Chaque sommet *i* ne peut accepter qu'un arc sortant vers un autre sommet *j*.

(3) Chaque sommet *j* ne peut accepter qu'un arc entrant d'un autre sommet *i*.

(4) Elimine la possibilité de sélection d'un arc permettant de former des circuits Hamiltoniens dans tout sous ensemble *P* de sommets [37].

II.6. Classification :

Globalement, le TSP est classé comme problème du voyageur de commerce symétrique (sTSP), asymétrique problème du voyageur de commerce (aTSP), et le problème du voyageur de commerce multiples (mTSP).

Une première particularité de ce problème se situe au niveau des caractéristiques du graphe. Effectivement, les arcs du graphe peuvent avoir un poids inégal selon le sens dans lequel on le parcourt, dans ce cas on dira que la matrice de ce problème est asymétrique. Dans le cas contraire, où la distance est la même peu importe le sens, alors cette dernière sera symétrique.

- sTSP :

Un problème est symétrique si la condition suivante est respectée : $c_{ij} = c_{ji}$ $\forall_i, j\ \forall_j$. Tel que c_{ij},c_{ji} et le coût entre le sommet i et j.

- aTsp :

Si $c_{ij} \neq c_{ji}$ pour au moins 2 sommet, le problème devient asymétrique.

- mTsp :

Le mTSP est défini comme suit: Dans un ensemble donné de nœuds, m commerçants situés à un nœud de dépôt. Les nœuds (villes) qui sont à visiter sont des nœuds intermédiaires. Ensuite, le mTSP consiste à trouver des visites pour les m commerçants, qui débutent et finissent au dépôt, de telle sorte que chaque nœud intermédiaire est visité une seule fois et le coût total de visites soit réduit au minimum.

Plusieurs variations de problème sont possibles : un ou plusieurs dépôts, nombre de commerçants (fixé), coût (fixé), calendrier (délai de visite des nœuds) [38].

II.7. Les applications de TSP :

- Forage de cartes de circuits imprimés

Une application directe de la TSP est dans le problème de perçage de circuits imprimés [39]. Pour connecter un conducteur sur une couche avec un conducteur sur une autre couche, ou de positionner les broches de circuits intégrés, les trous doivent être percés à travers la carte. Les trous peuvent être de tailles différentes. La machine doit percer tous les trous de même diamètre et changer la perceuse. L'objectif est de minimiser le temps Voyage pour la tête de la machine.

- Révision de moteurs à turbine à gaz

Plante et al. (1987) [40] ont rapporté cette application qui se produit lorsque les turbines à gaz d'aéronefs doivent être révisés. Pour garantir un écoulement de gaz uniforme à travers les turbines, il y a des ensembles d'aubes de guidage de tuyère situés à chaque étage de turbine. Toutes ces aubes ont des caractéristiques individuelles et le placement correct des aubes peut entraîner des avantages. Le problème de la mise en place des aubes de la meilleure façon possible peut être modélisé comme un TSP.

- Cristallographie aux rayons X

L'analyse de la structure des cristaux [41, 42] est une application importante de la TSP. Pour obtenir des informations sur un cristal, un détecteur mesure l'intensité de la réflexion de rayon X du cristal à partir de différentes positions. Par conséquent, le problème consiste à trouver une séquence qui minimise le temps de positionnement global. Cela conduit à un problème du voyageur de commerce.

- Le câblage d'ordinateur

Lenstra et Rinnooy Kan (1974) [43] ont signalé un cas particulier de la connexion des composants sur un ordinateur de bord. Les modules sont situés sur une carte d'ordinateur et un sous-ensemble donné de broches doit être raccordée. La condition est que pas plus de deux fils sont attachés à chaque broche. On a donc le problème de trouver un chemin Hamiltonien le plus court avec départ non spécifié.

- Conception de réseaux mondiaux d'arpentage du système de navigation par satellite

Une très récente et intéressante application de TSP qui est la conception d'un système mondial de navigation par satellite (GNSS) arpentage réseaux. Un GNSS est un système de satellite spatial qui fournit une couverture pour tous les endroits à travers le monde. Les récepteurs qui détermine les zones géographiques sont placés et coordonnés par une série de séances d'observation. Quand il y a plusieurs récepteurs ou plusieurs périodes de travail, le problème de trouver le meilleur ordre de sessions pour les récepteurs peut être formulé comme un TSP [44].

II.8. Méthodes de résolution

La résolution de TSP consiste à trouver la meilleure solution, définie comme la solution globalement optimale.

La résolution des problèmes combinatoires est assez délicate puisque le nombre fini de solutions réalisables croît généralement avec la taille du problème, ainsi que sa complexité. Cela a poussé les chercheurs à développer de nombreuses méthodes de résolution en recherche opérationnelle (RO) et en intelligence artificielle (IA).

Les algorithmes de résolution du TSP peuvent être répartis en deux classes :

- Les algorithmes exacts permettent de trouver la solution optimale, mais leur complexité est exponentielle.
- Les algorithmes heuristiques (d'approximation) obtiennent de bonnes solutions mais ne donnent aucune garantie sur l'optimalité de la solution trouvée.

II.8.1 Les méthodes de résolution exactes

Un algorithme exact permet de trouver la solution optimale. Or, cela exige un temps de calcul important puisque implicitement elle consiste à énumérer l'ensemble des solutions possibles. Ainsi, comme le temps de calcul risque d'augmenter exponentiellement avec la taille du problème, ces méthodes rencontrent des difficultés lorsque la taille du problème augmente.

II.8.1.1. La méthode séparation et évaluation (Branch and Bound)

L'algorithme de séparation et évaluation, plus connu sous son appellation anglaise Branch and Bound (B&B) [45], repose sur une méthode arborescente de recherche d'une solution optimale par séparations et évaluations, en représentant les états solutions par un arbre d'états, avec des nœuds, et des feuilles.

Le branch-and-bound est basé sur trois axes principaux :

- L'évaluation.
- La séparation.
- La stratégie de parcours.

- **L'évaluation**

Permet de réduire l'espace de recherche en éliminant quelques sous-ensembles qui ne contiennent pas la solution optimale en comparant le coût du nœud avec le coût de la solution optimale trouvé.

- **La séparation**

  La séparation consiste à diviser le problème en sous-problèmes. Ainsi, en résolvant tous les sous-problèmes et en gardant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial. Cela revient à construire un arbre permettant d'énumérer toutes les solutions.

- **La stratégie de parcours**
  - **La largeur d'abord :** Cette stratégie favorise les sommets les plus proches de la racine en faisant moins de séparations du problème initial. Elle est moins efficace que les deux autres stratégies présentées.
  - **La profondeur d'abord :** Cette stratégie avantage les sommets les plus éloignés de la racine (de profondeur la plus élevée) en appliquant plus de séparations au problème initial. Cette voie mène rapidement à une solution optimale en économisant la mémoire
  - **Le meilleur d'abord :** Cette stratégie consiste à explorer des sous-problèmes possédant la meilleure borne. Elle permet d'éviter l'exploration de tous les sous-problèmes qui possèdent une mauvaise évaluation par rapport à la valeur optimale.

Considérons le problème du voyageur de commerce afin d'expliquer la méthode de Branch&Bound.

Pour n villes v_0, v_1,…, v_n et les distances inter-villes c_{ij}, i, j ∈ {0, …, n}. Une solution de ce problème s'écrit donc sous la forme d'une liste de villes données dans l'ordre de la tournée, que l'on peut noter $\sigma(1),\ldots,\sigma$ (n) où σ est une permutation des n villes. On a donc clairement (n)! solutions possibles. Nous avons vu que l'explosion combinatoire d'une énumération la rend difficilement envisageable dans tous les cas.

Les méthodes de Branch&Bound essayent d'éviter d'explorer entièrement l'espace des solutions en utilisant l'évaluation et la stérilisation. Malheureusement, ces améliorations ne changent pas la complexité de ces méthodes qui restent de complexité exponentielle.

Prenons dans cet exemple 5 villes v_0, v_1, v_2, v_3 et v_4 et les distances inter-villes données par le tableau suivant :

	v_0	v_1	v_2	v_3	v_4
v_0	-	8	4	2	3
v_1	9	-	$\overline{7}$	1	6
v_2	3	$\overline{7}$	-	6	6
v_3	2	1	6	-	4
v_4	$\overline{7}$	6	6	2	-

Tableau II-1 : Table des distances

Ce qui aboutit en partie à l'arborescence d'énumération suivante :

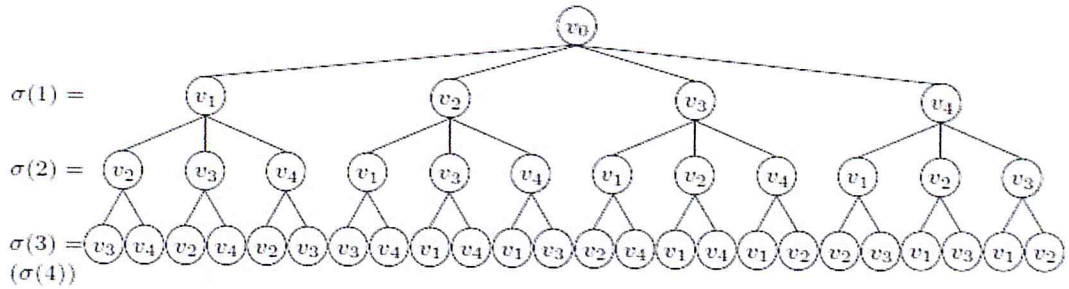


Figure II-1 Un arbre d'exploration de B&B [46]

Nous obtiendrons la solution au problème en explorant et évaluant tous les branches qui peuvent représenter un chemin optimale [46].

II.8.1.2. La méthode de coupes planes (Cutting-Plane)

La méthode de coupes planes a été développée par [47], elle est destinée à résoudre des problèmes d'optimisation combinatoire (POC) qui se formulent sous la forme d'un programme linéaire (PL) :

Min $\{ c^T x : Ax \geq b, x \in R^n \}$ ..... (1)

Dans le cas où (POC) est de grande taille pour le représenter explicitement en mémoire ou pour qu'il tienne dans une résolution de programmation linéaire, on utilise une technique qui consiste à enlever une partie de ces contraintes et de résoudre le problème relaxé (POCR). La solution optimale de (PL) est contenue dans l'ensemble de solutions réalisables de cette relaxation. Pour un problème de minimisation la solution optimale du problème (POCR) est inférieure ou égale à la solution optimale donnée par (POC).

Cette méthode consiste à résoudre un problème relaxé, et à ajouter itérativement des contraintes du problème initial. On définit une contrainte pour le problème de minimisation(1) par le couple (s, s_0) où $s \in R_n$ et $s_0 \in R$, cette contrainte est dite violée par la solution courante x si pour tout $y \in \{x : Ax \geq b\}$ on a $s^T\overline{x} < s_0$ et $s^T y \geq s_0$, on appelle alors ces contraintes des coupes planes. On arrête l'algorithme lorsqu'il n'y a plus de contraintes violées par la solution courante, on obtient ainsi une solution optimale pour le problème initial.

La méthode des coupes planes est peu performante mais sa performance est améliorée lorsqu'elle est combinée avec la méthode "Branch and Bound" [48].

II.8.1.3. Programmation dynamique et dominance

La structure d'un problème permet fréquemment d'éviter une exploration systématique comme l'effectue un algorithme de B&B. On peut considérer deux cas.

Lorsque les valeurs des solutions sont liées entre elles par une formule de récurrence basée sur le principe suivant. Supposons qu'une solution d'un problème est déterminé par une suite de décisions D_1, D2,…, D_p : l'hypothèse dite de la programmation dynamique est qu'une prise de décisions optimales D_1,…,D_p est telle que, pour tout entier k = 1,…p - 1, les décisions D_{k+1},…, D_p doivent être optimales. Cette hypothèse est très forte car elle a pour conséquence de donner un ordre à l'exploration des solutions en éliminant l'exploration de certaines. De plus, sous certaines conditions à prouver, elle permet même de montrer qu'il n'y a qu'un nombre polynomial de sous-problèmes (appelés alors états à explorer.

On peut citer comme exemple l'algorithme de Dijkstra pour les plus courts chemins ou la résolution des cas polynomiaux du problème du voyageur de commerce.

Lorsque les solutions ont des structures fortement liées entre elles, on peut détecter des 'dominances' entre solutions : si dans une arborescence, on peut montrer que tout sous-problème descendant d'un sous-problème a est au moins aussi bon que tout descendant d'un sous-problème b : on dit que a domine b. Dans ce cas, il suffit d'explorer a : on dit que a tue b (cela revient à stériliser b dès que a est explorer par exemple).

Ces dominances se rencontrent dans les problèmes à forte structure, par exemple des problèmes d'ordonnancement ou d'optimisation sur les graphes, où les solutions ont des points ou des parties en communs [46].

II.8.2 Les méthodes heuristique

Si les méthodes de résolution exactes permettent d'obtenir une solution dont l'optimalité est garantie, dans certaines situations, on peut cependant chercher des solutions de bonne qualité, sans garantie d'optimalité, mais au profit d'un temps de calcul plus réduit. Pour cela, On applique des méthodes appelées heuristiques, adaptées à chaque problème traité, avec cependant l'inconvénient de ne disposer en retour d'aucune information sur la qualité des solutions obtenues.

Les heuristiques ou les méta-heuristiques exploitent généralement des processus aléatoires dans l'exploration de l'espace de recherche pour faire face à l'explosion combinatoire engendré par l'utilisation des méthodes exactes. En plus de cette base stochastique, les méta-heuristiques sont

le plus souvent itératives, ainsi le même processus de recherche est répété lors de la résolution. Leur principal intérêt provient justement de leur capacité à éviter les minima locaux en admettant une dégradation de la fonction objective au cours de leur progression [48].

II.8.2.1 Les méthodes de recherche locale

La recherche locale est la base de nombreuses méthodes méta-heuristiques pour des problèmes d'optimisation combinatoire, dans la méthode recherche locale l'ensemble S définit l'ensemble des points pouvant être visités durant la recherche. La structure de voisinage N donne les règles de déplacement dans l'espace de recherche. La fonction objective f induit une topologie sur l'espace de recherche.

a) La méthode descente

Cette méthode de recherche locale est l'une des plus simples de la littérature, elle est également appelée hill climbing dans les problèmes de maximisation. Son principe consiste, à partir d'une solution initiale, à choisir à chaque itération un voisin qui améliore strictement la fonction objective.

Il existe plusieurs moyens de choisir ce voisin, soit par le choix aléatoire d'un voisin parmi ceux qui améliorent la solution courante (first improvement), soit en choisissant le meilleur voisin qui améliore la solution courante (best improvement). Dans tous les cas, le critère d'arrêt est atteint lorsque plus aucune solution voisine n'améliore la solution courante.

La méthode descente peut être décrite comme suit :

1. Solution initiale s ;
2. **Répéter** :
3. Choisir s_0 dans un voisinage V(s) de s ;
4. Si $f(s_0) < f(s)$ alors $S := S_0$;
5. **jusqu'à** ce que $f(s_0) \geq f(s), \forall S_0 \in V(s)$.

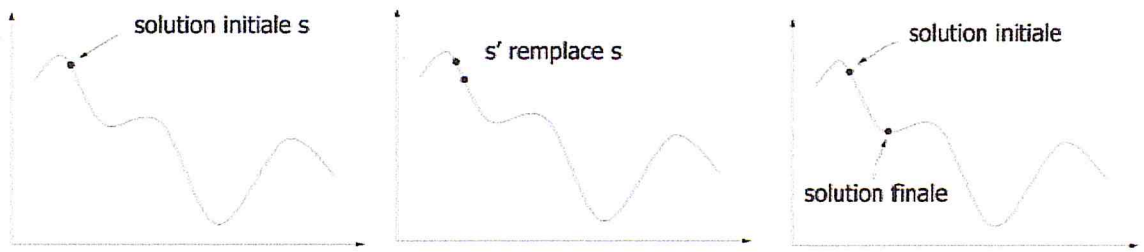


Figure II-2 : Evolution d'une solution dans la méthode de descente [48]

La figure II-2 présente l'évolution d'une solution dans la méthode de descente.

L'inconvénient majeur de la méthode de descente est son arrêt au premier minimum local rencontré. Pour améliorer les résultats, on peut lancer plusieurs fois l'algorithme en partant d'un jeu de solutions initiales différentes, mais la performance de cette technique décroît rapidement [48].

b) Recherche Tabou **(Tabu Search)**

La recherche tabou (TS) est une méthode de recherche locale combinée avec un ensemble de techniques permettant d'éviter d'être piégé dans un minimum local ou la répétition d'un cycle. La recherche tabou a été introduite principalement par Glover [49], Hansen [50], Glover et Laguna dans [51].Cette méthode a montré une grande efficacité pour la résolution des problèmes d'optimisation difficiles. En effet, à partir d'une solution initiale s dans un ensemble de solutions local S, des sous-ensembles de solution N(s) appartenant au voisinage S sont générés. Par l'intermédiaire de la fonction d'évaluation nous retenons la solution qui améliore la valeur de f, choisie parmi l'ensemble de solutions voisines N(s).
La mémoire, appelée liste tabou, contient la liste des solutions (ou des zones) récemment visitées, et permet d'éviter de cycler et de retomber en permanence dans l'optimum local duquel on vient de sortir. Ce procédé simple permet alors de sortir de l'optimum local et de se diriger vers d'autres régions de l'espace des solutions. Pour résumer, on peut dire qu'à chaque itération, on choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction objective [48].
Différentes améliorations de cette méthode ont été mises au point [52].

La méthode tabou exige une gestion de la mémoire de plus en plus lourde en mettant des stratégies de mémorisation complexe. L'efficacité de la méthode tabou offre une utilisation dans plusieurs problèmes d'optimisation combinatoire classiques tels que le problème de voyageur de commerce, le problème d'ordonnancement, le problème de tournées de véhicules, etc. [48].

c) Recuit simulé (Simulated annealing)

Le recuit simulé (SA) a été introduit par Kirkpatrik et al. (1983) [53] et Cerný (1985) [54] comme une méthode de recherche locale normale, utilisant une stratégie pour éviter les minima locaux. Cette méta-heuristique est basée sur une technique utilisée depuis longtemps par les métallurgistes qui, pour obtenir un alliage sans défaut, faisant alterner les cycles de réchauffage (ou de recuit) et de refroidissement lent des métaux. Le recuit simulé s'appuie sur des travaux faites par Metropolis et al. (1953) [55], qui ont pu décrire l'évolution d'un système en thermodynamique.

Ce processus utilisé en métallurgie pour améliorer la qualité d'un solide cherche un état d'énergie minimale qui correspond à une structure stable du solide.

En partant d'une haute température à laquelle le solide est devenu liquide, la phase de refroidissement conduit la matière liquide à retrouver sa forme solide par une diminution progressive de la température. Chaque température est maintenue jusqu'à ce que la matière trouve un équilibre thermodynamique. Quand la température tend vers zéro, seules les transitions d'un état à un état d'énergie plus faible sont possibles [56].

Le principe du recuit simulé est de parcourir de manière itérative l'espace des solutions. On part avec une solution notée s0 initialement générée de manière aléatoire dont correspondent une énergie initiale E_0, et une température initiale T_0 généralement élevée. A chaque itération de l'algorithme, un changement élémentaire est effectué sur la solution, cette modification fait varier l'énergie du système DE. Si cette variation est négative (la nouvelle solution améliore la fonction objective, et permet de diminuer l'énergie du système), elle est acceptée. Si la solution trouvée est moins bonne que la précédente alors elle sera acceptée avec une probabilité P calculée suivant la distribution de Boltzmann suivante :

$$P(E, T) = e^{-\frac{\Delta E}{T}} \text{ .....(1)}$$

En fonction du critère de Metropolis, un nombre $\epsilon \in 2\ [0, 1]$ est comparé à la probabilité $p=e^{-\frac{\Delta E}{T}}$.

Si $p \leq \epsilon$ la nouvelle solution est acceptée.

La méthode du recuit simulé a l'avantage d'être :

- souple vis-à-vis des évolutions du problème et facile à implémenter,
- Contrairement aux méthodes de descente, SA évite le piège des optima locaux,

Mais de nombreux tests sont nécessaires pour trouver les bons paramètres, et les difficultés lors de la définition des voisinages permettant un calcul efficace de ΔE [48].

d) Récapitulatif des recherches locales

L'élément qui caractérise une recherche locale est le choix de la solution voisine s' dans le voisinage V(s).

- Descente :

s' est LA solution voisine la plus performante $f(s') \leq f(s)$ et $\forall s'' \in V(s)\ f(s') \leq f(s'')$.

- Recuit simulé :

s' est choisie au hasard :

- acceptée si plus performante que s'.
- acceptée si moins performante que s avec une probabilité donnée.

- Recherche tabou :

S' est l'une des meilleures solutions voisines de s, n'appartenant pas à la liste tabou [57].

Avantages

- Méthodes rapides (et temps paramétrable).
- Faciles à implémenter.
- Donnent souvent de bonnes solutions.
- Fonctionnement intuitif.

Inconvénients

- Manque de modélisation mathématiques (chaque cas est différent : il faut adapter la recherche à chaque problème particulier).
- Difficile à paramétrer.
- Aucune évaluation de la distance à l'optimum (pas une approximation, ne trouve au pire qu'un optimum local qui n'a rien à voir).

II.8.2.2. Algorithme de Lin-Kernighan

L'algorithme commence avec une tournée admissible donnée ; cherche ensuite dans le voisinage de la solution courante défini par l'opération *λ-opt move* toute tournée améliorant la configuration courante. A chaque étape de l'itération, l'algorithme examine, pour des valeurs croissantes de λ (à partir de 2) si l'échange de λ liens produit une tournée plus courte. L'algorithme continue ainsi jusqu'à ce qu'aucune amélioration ne soit plus possible. L'opération *λ-opt move* consiste à supprimer λ liens (arêtes) et à reconnecter les segments restants par de nouveaux liens, en renversant si possible le sens de parcours d'un ou de plusieurs de ces segments. Plus la valeur de λ est grande, plus la solution finale est proche de l'optimum et plus le temps d'exécution devient élevé. En général on utilise des valeurs entières de $\lambda \in \{2, 3, 4, 5\}$. La Figure II-3 illustre l'opération *2-opt move* [58].

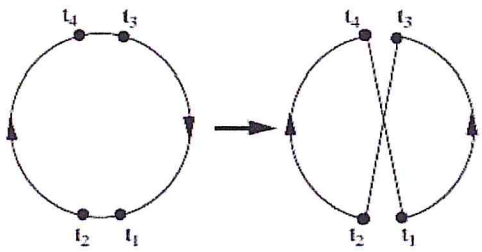


Figure II-3 : 2-opt move [59]

Les résultats de cet algorithme sont très prometteurs en optimalité, car, en appliquant toutes les étapes d'amélioration de solution, on est presque à 0% de taux d'erreur par rapport à

la solution optimale, pour des échantillons de petites et moyennes tailles. Par contre, les temps d'exécution sont peu compétitifs [59].

II.8.2.3. Les méthodes Gloutonne

Il s'agit d'algorithmes cherchant à résoudre le problème en un certain nombre d'étapes. Les étapes fixent successivement la valeur de toutes les variables du problème sans jamais reconsidérer les choix des étapes précédentes. On dit qu'ils "avalent" les variables (d'où le nom "glouton") [60].

Un algorithme glouton (*greedy algorithm* en anglais) est un algorithme qui suit le principe de faire, étape par étape, un choix optimum local, dans l'espoir d'obtenir un résultat optimum global. Dans le TSP une heuristique gloutonne construit une seule solution, par une suite de décisions définitives sans retour arrière, parmi ces méthodes on cite le plus proche voisin, la plus proche insertion, la plus lointaine insertion et la meilleure insertion.

a) La méthode du plus proche voisin

En partant d'un sommet quelconque et à chacune des (n - 1) itérations on relie le dernier sommet atteint au sommet le plus proche au sens coût, puis on relie finalement le dernier sommet au premier sommet choisi.

b) Méthodes d'insertion d'arc

Dans cette méthode, on part d'un cycle réduit à une boucle au départ, à chaque itération on choisit un sommet libre j puis on cherche la position d'insertion i et j de cycle qui minimise l'augmentation totale des coûts :

- dans *la plus lointaine insertion* j est le sommet libre le plus loin du cycle au sens des coûts;
- dans l*a plus proche insertion* j est le plus proche du cycle;
- enfin dans la *meilleure insertion* on teste tous les sommets j non encore insérés et on choisit celui qui donne la plus faible augmentation du coût.

Les méthodes gloutonnes ont principalement une grande simplicité de mise en œuvre ainsi qu'un temps d'exécution raisonnable. En contrepartie, ils peuvent parfois aboutir à une solution relativement éloignée de la solution optimale. La qualité de la solution fournie dépend énormément (mais pas entièrement) des fonctions Choisi et Fixé. On essaye habituellement plusieurs variantes de ces fonctions pour équilibrer au mieux la qualité de la solution et temps d'exécution [60].

II.8.2.4. Les Algorithmes génétiques

Les algorithmes évolutifs sont une famille d'algorithmes issus de la théorie de l'évolution par la sélection naturelle, énoncée par Charles Darwin en 1859.

L'évolution naturelle a en effet permis de créer des systèmes biologiques très complexes. Le principe fondamental étant que les individus les mieux adaptés à leur environnement survivent et peuvent se reproduire, laissant une descendance qui transmettra leurs gènes. Cette conclusion étant évidemment impossible si tous les individus avaient le même bagage génétique, on suppose donc que des variations non dirigées (mutations) du matériel génétique des espèces peuvent apparaître aléatoirement.

La clef étant l'adaptation des individus face à la pression de l'environnement, l'analogie avec l'optimisation devient claire. Cette adaptation peut alors être assimilée à une optimisation des individus afin qu'ils soient de mieux en mieux adaptés à leur environnement, au fur et à mesure des générations (correspondant aux itérations de l'algorithme). Nous pourrons alors définir les algorithmes évolutionnaires comme des méthodes faisant évoluer un ensemble de solutions appelé "population". Les solutions, appelées "individus", sont représentées par leur génotype, qui s'exprime sous la forme d'un phénotype. Afin d'évaluer la performance d'un individu, on associe au phénotype la valeur de la fonction objectif (ou fonction d'évaluation). Celle-ci se distingue de la fonction *fitness* qui représente l'évaluation d'un individu quant à sa survie dans la population. Cette méthode permet de s'assurer que les individus performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population. Dans le cadre des méta-heuristiques, l'exploration de l'espace de recherche est alors réalisé par les opérateurs de mutation qui assure la diversification des individus de la population (et donc des solutions). L'exploitation, quant à elle, est assurée par les opérateurs de croisement, qui recombinent les solutions, afin de les améliorer en conservant leurs meilleures caractéristiques. Dans les années 60 et 70, quatre grandes écoles utilisant ce principe général de façon différente furent développées indépendamment. Ces différentes familles d'algorithmes évolutionnaires ne diffèrent que par la structure du génotype des individus ou par les opérateurs utilisés.

Les principales fonctions des algorithmes génétiques :

- Fonction d'adaptation

Il s'agit de caractériser l'adaptation d'un individu au critère visé.

- Génération de nouveaux individus

Le principe consiste à générer un nouvel individu à partir d'un individu (par mutation), de deux individus (par l'hybridation) de la population père ou par la reproduction.

- La sélection

La sélection intervient à deux stades : lors du choix des reproducteurs et lors de la fabrication de la population. Différentes méthodes de sélection existent chacune avec ses critères de sélection [48].

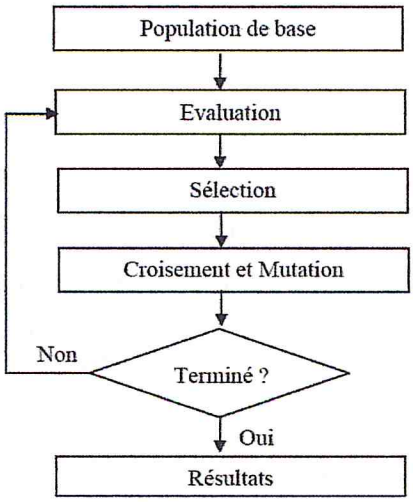


Figure II-4 : Les étapes d'un algorithme génétique [48]

II.8.2.5. Les Algorithmes de colonies de fourmis

Ces méthodes d'optimisation proviennent d'analogies avec des phénomènes biologiques naturels. La particularité des algorithmes issus de l'intelligence en essaim (ou swarm intelligence) est d'utiliser une population d'agents (alors appelée essaim) au lieu d'une simple population de solutions. Chaque agent de l'essaim pourra agir indépendamment des autres, il aura son propre système de décision, gouverné par un ensemble de règles. Un tel système est appelé système multi-agent, et aboutit à l'émergence d'un comportement pour l'essaim entier (ici le comportement sera la convergence vers une solution du problème d'optimisation). Les algorithmes de ce type les plus connus sont les méthodes de colonies de fourmis et d'optimisation par essaim particulaire. D'autres méthodes plus récentes commencent néanmoins à percer, on peut notamment citer les colonies d'abeilles artificielles [52].
Nous étudions plus en détail les algorithmes de colonie de fourmis ou Ant Colony Optimization (ACO).

L'idée initiale de cette méthode provient de l'observation du comportement des fourmis lorsqu'elles cherchent de la nourriture [61]. En effet, celles-ci parviennent à trouver le chemin le plus court entre leur nid et une source de nourriture, sans pour autant avoir des capacités cognitives individuelles très développées. Après cette étude, un principe étonnant fut révélé : les fourmis communiquent indirectement via leur environnement (on parle alors de

L'algorithme général est relativement simple, et repose sur un ensemble de fourmis, chacune parcourant un trajet parmi ceux possibles. À chaque étape, la fourmi choisit de passer d'une ville à une autre en fonction de quelques règles :

1) elle ne peut visiter qu'une fois chaque ville.

2) plus une ville est loin, moins elle a de chance d'être choisie (c'est la « visibilité »).

3) plus l'intensité de la piste de phéromone disposée sur l'arête entre deux villes est grande, plus le trajet aura de chance d'être choisi.

4) une fois son trajet terminé, la fourmi dépose, sur l'ensemble des arêtes parcourues, plus de phéromones si le trajet est court.

5) les pistes de phéromones s'évaporent à chaque itération en faisant disparaître les mauvaises solutions.

L'ACO est parfait pour les problèmes basée sur des graphes et a une grande adaptabilité.
Mais d'autre part un état bloquant peut arriver, ne s'applique pas à tous type de problèmes et il a un temps d'exécution parfois long [63].
Nous avons vu dans ce passage les différents algorithmes proposé pour résoudre le TSP. Les algorithmes exacte qui donnent la solution la plus optimale mais avec un temps d'exécution exponentiel.
Pour améliorer le temps d'exécution nous avons vu les méta-heuristiques qui fournissent une solution proche de l'optimale (qui satisfait le problème). D'où nous avons les algorithmes de recherche locale, constructive (Glouton), génétique et intelligence de l'essaim. Chacun d'eux ayant des avantages et des inconvénients.
Différentes mesures seront appliqué sur les données d'entré pour avoir les poids entre les sommets (dimensions). La matrice résultante peut être symétrique ou asymétrique.
Le premier critère est de choisir des algorithmes capables de résoudre les deux de TSP (sTSP et aTSP). Un autre critère est la performance (rapidité et qualité de la solution).
Pour résoudre notre problème nous utiliserons un algorithme exact, l'algorithme de Branch & Bound et deux méta-heuristiques qui sont l'algorithme de plus proche voisin de la famille des Glouton et l'algorithme de colonie de fourmis.
Notre principal choix est porté sur l'algorithme de clonie de fourmis qui est bon pour les problèmes basés sur des graphes, est bien adapté au TSP. Ces résultats sont très bons avec taux d'erreur bas [63].
Nous utilisons aussi l'algorithme de plus proche voisin pour sa rapidité et la simplicité d'implémentation [60].

L'algorithme de Branch & Bound fournit la solution la plus optimale ce qui nous permettra d'évalué le résultat des autres approches utilisé.

II.9. Conclusion

Ce chapitre nous a permis de connaitre le TSP, ces aspects et les méthodes de résolution afin de pouvoir l'adapter pour l'ordonnancement des dimensions sur le ParCoors. Pour résoudre notre problème, nous avons choisi trois approches différentes. Ce choix multiple de méthodes nous permettra d'avoir différents ordres des dimensions pour comparer la performance de chacune et voir son impact sur l'affichage, l'interprétation et l'extraction des connaissances.

Chapitre

III. Amélioration de l'analyse visuelle

III.1. Introduction

Une bonne visualisation révèle clairement la structure dans les données et peut donc aider le visionneur à mieux identifier les tendances et de détecter les valeurs aberrantes. L'encombrement visuel, d'autre part, est caractérisé par des entités visuelles surpeuplées et désordonnés qui obscurcissent la structure de la représentation visuelle. En d'autres termes, cet aspect visuel est l'opposé des formes visuelles, il correspond à l'ensemble des facteurs qui interfèrent avec la détection de forme. Ce problème est certainement non souhaitable, car elle entrave la compréhension des visionneurs du contenu des graphes mais lorsque le nombre de dimensions ou de données croît, l'affichage exposé devient chargé [7].

Puisque l'encombrement cache des données importantes et des structures dans la représentation. Il est nécessaire de le réduire. En ParCoords, l'encombrement est dû aux valeurs aberrantes, au croisement des poly-lignes et le grand nombre de données. L'ordonnancement des dimensions est la technique la plus populaire pour la réduction de l'encombrement et finalement l'amélioration de l'affichage.

III.2. Intégration de l'amélioration dans le processus de visualisation

Card, Mackinlay & Shneiderman (1999) ont introduit un modèle de référence pour la visualisation d'information, qui offre une vue globale sur le processus de visualisation d'information.

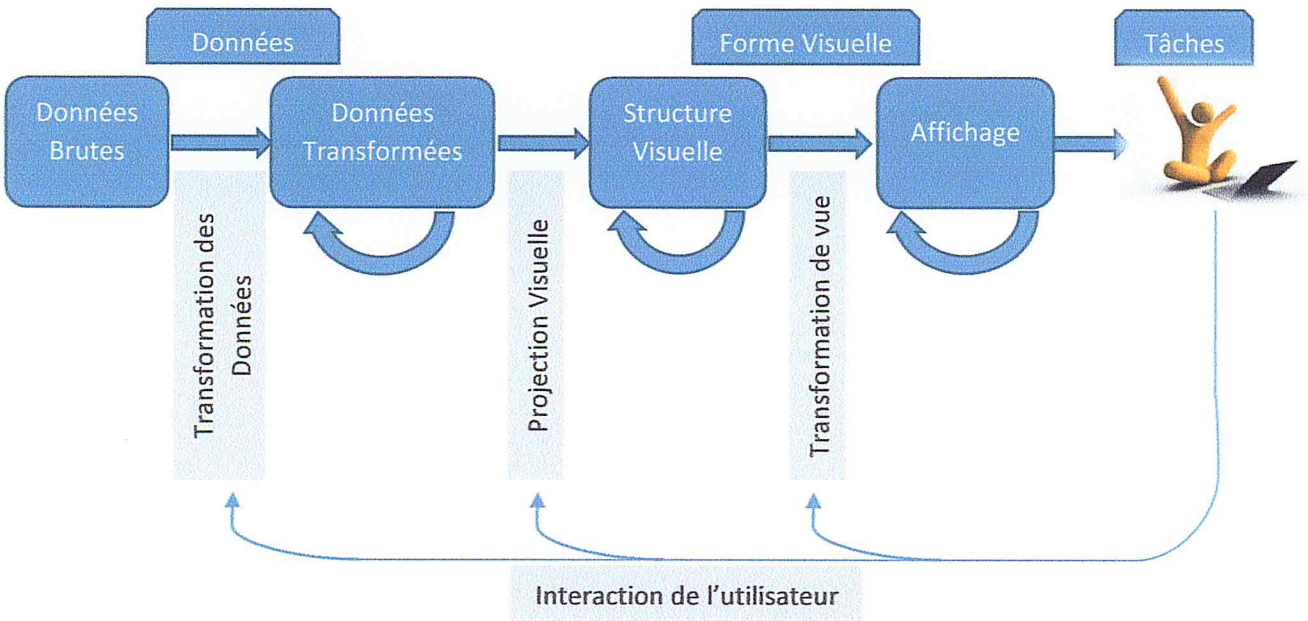


Figure III-1 : Processus de visualisation [13]

Le modèle (figure III-1) prend en charge les données brutes, qui existent dans un format propriétaire. Pour arriver à une visualisation de ces données, les données doivent d'abord subir une série de transformations. Les transformations de données comprennent le filtrage des données brutes, calcul des données dérivées, ainsi que la normalisation des données. Ces étapes se traduisent par un ensemble de données transformées dans une structure unifiée. La transformation visuelle projette les données transformées sur une structure visuelle correspondante. De cette structure visuelle, un ensemble d'affichage peut être généré, qui permet aux utilisateurs de naviguer à travers l'écran. Les flèches cycliques dans le diagramme se réfèrent au fait que les processus impliqués dans des étapes distinctes sont de nature itérative et peuvent se produire plusieurs fois avant l'étape qui suit.

Type de données : Ce sont les types définis par D.Keim (2002) [1] qui sont unidimensionnelles, bidimensionnelles, multidimensionnelles, textes et hypertextes, hiérarchies et graphiques, algorithmes et logiciels.

Représentations visuelle : Les structures visuelles pour exprimer les données (informations) selon les techniques présentées dans le chapitre I.

Transformation de l'affichage : Le processus de présentation visuelle résulte en des structures graphiques qui représentent des informations. Dans une dernière étape, l'affichage construit ces structures graphiques et les rend accessibles à l'observateur humain.
Les transformations spécifient des paramètres graphiques qui influencent l'affichage comme la position, l'échelle et l'écrêtage. Variantes transformations peuvent révéler plus d'informations à partir d'une seule et même structure graphique.

Tâches : Shneiderman (1996) a énuméré sept tâches que les utilisateurs peuvent effectuer sur les données : vue d'ensemble, zoom, filtre, détails à la demande, rapport, histoire, et l'extraction.

Interaction : L'interaction humaine complète le processus de visualisation. L'utilisateur guide le processus de transformation aux différents stades. Il peut ajuster son affichage, modifier la structure visuelle, ou même affecter la transformation de données [13].

Dans notre travail nous adaptons ce modèle en prenant en considération la partie d'amélioration de l'affichage. Cette partie concerne l'ordonnancement automatique des dimensions.

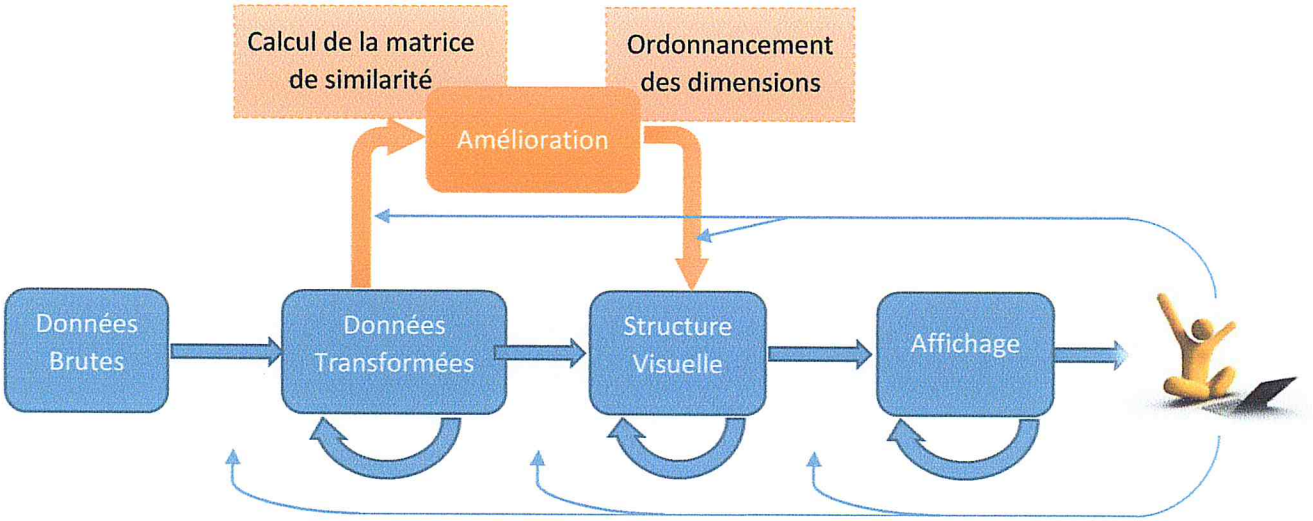


Figure III-2 : Processus de visualisation en ajoutant l'amélioration

Notre processus (figure III-2) commence par l'importation de données depuis des fichiers plats (.csv, .txt ou .xml). Ces données sont remplies dans des tables et peuvent être classées (cluster, class). Chaque ligne d'une table représente un élément et les colonnes, les dimensions (attributs) des données.

Avant leur utilisation une étape de prétraitement est nécessaire. Cette opération aura comme sortie la table qui sert à la projection sur le graphe (ParCoords) et une première représentation visuelle est affichée à l'utilisateur et les dimensions sont arrangées dans l'ordre par défaut.

Pour ordonner les dimensions de ParCoords, une matrice de similarité est calculée à partir des données transformées en appliquant une mesure choisie par l'utilisateur (Minkovski, Person, Spearman, Entropie).

Un algorithme d'ordonnancement est ensuite appliqué sur la matrice résultante pour calculer le plus court chemin qui relie les dimensions. Une autre présentation visuelle est affichée avec le nouvel ordre trouvé pour réduire l'encombrement aperçu sur le premier affichage.

Finalement, le visionneur utilise les outils d'interaction avec le graphe afin de mieux comprendre la représentation, analyser les données, interpréter les résultats et extraire des connaissances.

Dans la partie suivante, nous détaillons notre processus de visualisation (ClusterViz).

III.2.1 Intégration des données

Lors de l'importation des données du fichier d'entrée, ces dernières ont besoin d'un prétraitement pour le bon déroulement du processus. Il consiste en :

- Filtrage : s'abstenir d'afficher les dimensions qui n'ont pas de valeur commune entre les données par exemple : Les noms de voiture (Chaque voiture a un nom propre à elle) ce qui fait de la dimension un mauvais axe d'analyse. Les dimensions qui n'ont qu'une seule valeur sont filtrées aussi.
- Traitement des données NULL : Remplir les champs vide avec la moyenne de la dimension à partir des autres données.
- Choix de la dimension de regroupement : La dimension dont laquelle les données ont la même valeur seront regroupées dans des clusters.

III.2.2 Génération des Structures Visuelles

La partie précédente se termine avec des données prêtes à l'utilisation. La construction de ParCoords se fait en plaçant en parallèle les dimensions. En haut des axes, sont placés les noms des dimensions contenues dans la première ligne de la table. Les valeurs maximum et minimum sont mises aux bornes supérieur et inférieur des axes. Chaque autre ligne est une donnée avec ces différentes valeurs.

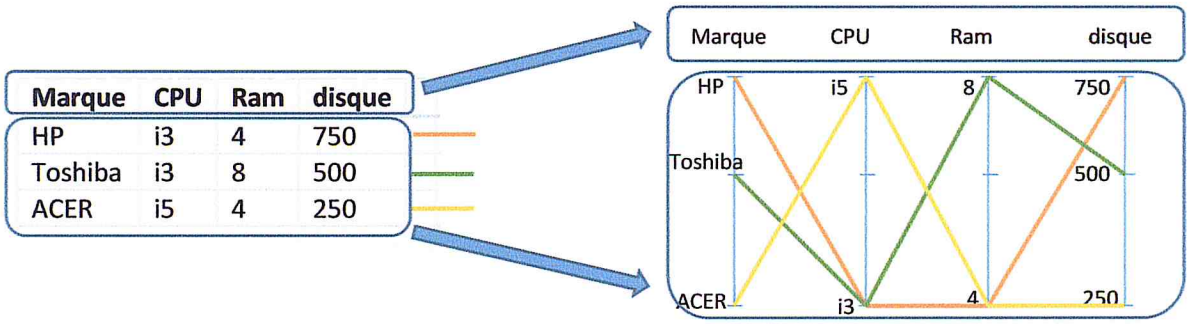


Figure III-3 : Construction des éléments de ParCoords

III.2.3 Génération des vues (Affichage)

Les structures graphiques (axes, poly-lignes, polygones) avec leur propriétés (couleur, forme, taille) construites seront affichées à l'utilisateur qui les visualise et interagi directement avec le graphe généré. La distance entre les axes dépend de leur nombre, ainsi que l'échelle dans laquelle les valeurs sont affichées.

III.2.4 Tâches de l'analyste

La tâche de l'utilisateur est l'exploration visuelle des données dans le but de trouver des informations de valeur, cachées dans les graphes et extraire les connaissances. Elle suit généralement un processus en trois étapes :

Aperçu premier, le zoom et le filtre, puis détails à la demande.

Tout d'abord, l'utilisateur a besoin d'obtenir une vue d'ensemble des données (les poly-lignes ou les polygones). Dans le graphe, il identifie par la suite les modèles intéressants (les

fromes ayant un comportement similaire ou différent) et met l'accent sur un ou plusieurs d'entre eux. Pour analyser les modèles, l'utilisateur doit effectuer une analyse approfondie et accéder aux détails des données.

Dans notre travail, le ParCoords fournit non seulement une technique de visualisation de base pour chacune des trois étapes, mais aussi montre le passage entre les étapes. Dans la phase d'exploration, l'analyste de données utilisera de nombreux graphiques qui peuvent révéler des caractéristiques très intéressantes et importantes.

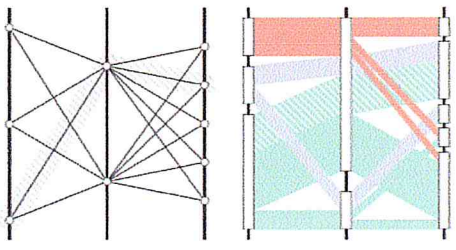
a) La visualisation des données

Selon les types de données visualisées, nous utilisons pour leur affichage différent graphes. Nous nous intéressons dans notre travail aux données multidimensionnelles. Où nous distinguons différent types :

- Quantitatif
  - Données continue : valeurs réelles (poids, taille, âge, glycémie, ...).
  - Données discrètes : valeurs isolées (échèle d'âge : 15, 20, 25).
- Qualitatif
  - Ordinale (Semi-Qualitatif) : avec relation d'ordre (A,…, Z, I, II, III, IV, …).
  - Données catégorielle : Nominale (groupes sanguins : A/B/AB/O, homme/femme, …).

Les valeurs quantitatives sont bien comprises lors de la visualisation sur ParCoords. Mais les dimensions qualitatives sont généralement des dimensions de données qui contiennent seulement un petit nombre de valeurs différentes, qui ont une signification particulière. Dans le cas ou plusieurs données ont la même valeur sur deux ou plusieurs dimensions, leur représentation est superposée (plusieurs poly-lignes chevauchées) en conséquence, des informations sont perdues. Pour éviter les pertes d'informations, nous ajoutons aux différentes approches de représentation, les Ensembles parallèles (Parallel Sets) [64].

Nous utilisons Parrallel Sets (ParSets) pour la représentation des données catégorielles.



(a) (b)

Figure III-4 : Le passage de ParCoords vers ParSets [64]

ParSets adopte les avantages de deux techniques de visualisation, la mise en page flexible de ParCoords, et l'affichage des fréquences en tant que représentants des catégories. Il partage la structure de ParCoords, mais les points d'intersection (figure III-4-a) sont remplacés par un ensemble de

rectangles qui représentent les catégories (figure III-4-b). Ces rectangles sont mis à l'échelle en fonction des fréquences des catégories correspondantes. Il donne une meilleure représentation des données catégorielles et garde l'information statistique.

Afin d'avoir une bonne exploration des données, l'utilisateur a la possibilté de représenter les données sur les deux graphes (ParCoords et ParSets).

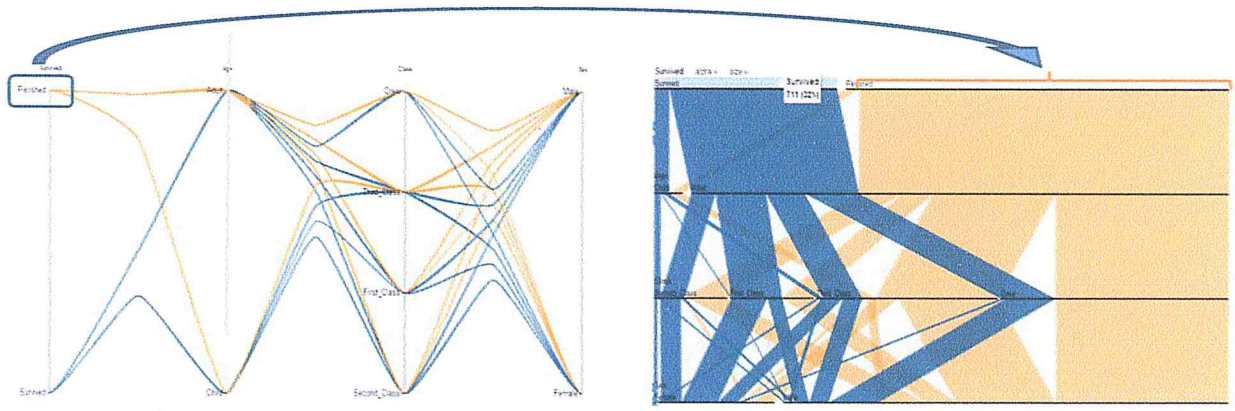


Figure III-5 : Présentation des données sur ParCoords et ParSets

b) Visualisation par cluster

L'interprétation des données et de l'extraction des connaissances devient difficile quand les ensembles de données sont trop grands. Afin de faciliter l'analyse visuelle, l'utilisateur peut balancer vers la visualisation des clusters des données.

Pour représenter les clusters, nous avons utilisé deux techniques :

- La coloration : Les données appartenant au même cluster auront la même couleur,
- Les lignes courbées : Cette méthode est caractérisée par deux fonctions. L'attraction, les données de même cluster s'attire entre elles, et une fonction de niveau d'inclinaisons des courbes. L'inclinaison des poly-lignes réduit l'encombrement à l'affichage et facilite l'interprétation,
- La représentation par MinMax : Les valeurs minimum et maximum de chaque cluster sont utilisées pour dessiner les polygones représentant les clusters.

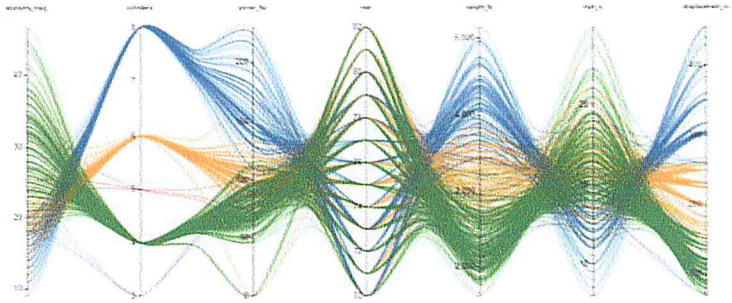


Figure III-6 : Application de coloriage et les courbes sur

III.2.5 Amélioration

L'étape d'ordonnancement et amélioration de l'affichage est composée de deux phases. Pour pouvoir appliquer l'ordonnancement sur les dimensions, une matrice de similarité qui contient les poids entre eux doit être calculée.

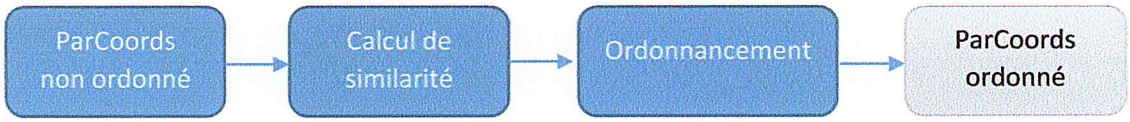


Figure III-7 : Processus d'amélioration

a) Calcul de la matrice de similarité

La corrélation est l'association ou l'interdépendance entre les axes, utilisée pour savoir s'il y a ou non une relation entre eux [14]. Pour calculer la similarité (distance) entre les dimensions nous avons appliqué différentes mesures [27].

Avant de calculer la similarité entre les dimensions, les données doivent être transformées et normalisées pour :

- Pouvoir utiliser les mesures sur les données quantitatives,
- éviter que l'incompatibilité des unités de mesures entre les variables affecte les résultats.

b) Transformation des données catégorielles

La dimension contenant des données quantitatives est ordonnée et les données nominales ou ordinales sont remplacées par des données numériques successivement.

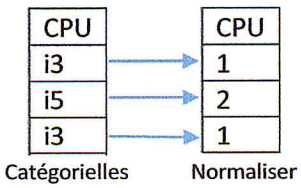


Figure III-8 : Transformation des données

c) La normalisation des données

Pour empêcher que la grande différence des valeurs de dimensions mène à un résultat faux, nous appliquons sur eux la normalisation suivante :

$$\frac{Valeur - Valeur\ Minimum\ de\ Dimension}{Valeur\ Maximum - Valeur\ Munimum} \ldots\ldots\ldots (1)$$

Les données seront comprises dans le même intervalle [0,1].

d) Les mesures de similarité

➢ La corrélation de Pearson

Informellement, la corrélation répond à la question "si j'augmente (ou diminue) x, est-ce que y augmentera (ou diminuera), et de combien ?". Formellement, elle mesure la dépendance linéaire de deux variables quelconques. Ses valeurs varient de -1 à 1 [65].

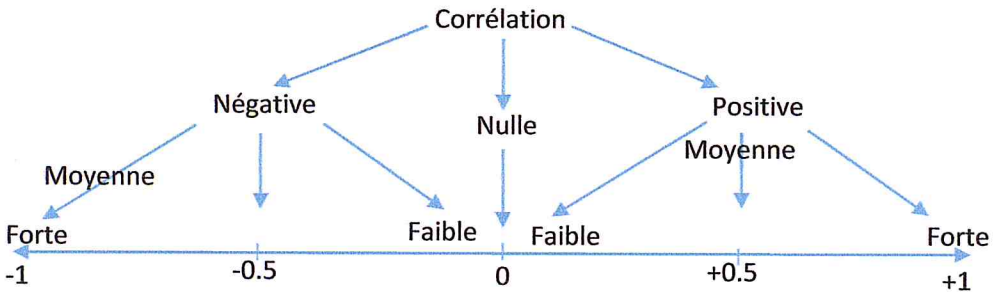


Figure III-9 : interprétation de la corrélation [66]

- Si r est proche de 0, il n'y a pas de relation.
- Si r est proche de -1, il existe une forte relation négative.
- Si r est proche de 1, il existe une forte relation positive [66].

La mesure de corrélation la plus utilisée est le coefficient de corrélation de Pearson. Ce coefficient permet de détecter la présence ou l'absence d'une relation linéaire entre deux caractères quantitatifs continus. Pour calculer ce coefficient il faut tout d'abord calculer la covariance. La covariance est la moyenne du produit des écarts à la moyenne.

$$\rho(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} * \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \dots\dots\dots (1)$$

Où $\bar{x} = \frac{1}{N}\sum_{i=1}^{n} x_i$ est la moyenne de X et $\bar{y} = \frac{1}{N}\sum_{i=1}^{n} y_i$ la moyenne de Y.

Le coefficient de Pearson n'est applicable que pour mesurer la relation entre deux variables quantitatives ne contenant pas des valeurs de grande différence. Dans le cas contraire, l'emploi de ce coefficient peut aboutir à des conclusions erronées sur la présence ou l'absence d'une relation [66].

➢ La corrélation de Spearman

Le coefficient de corrélation de Spearman (appelé coefficient de rang) examine s'il existe une relation entre le rang des observations pour deux données, ce qui permet de détecter

l'existence de relations monotones (croissantes ou décroissantes). On notera également qu'il est préférable au coefficient de Pearson lorsque les données sont dissymétriques et/ou comportent des valeurs de grande différence. Contrairement à la corrélation de Pearson, la condition de bon déroulement (type et valeur des données) n'est pas nécessaire (utilisation des rangs et non des valeurs).

Le coefficient de Spearman est fondé sur l'étude de la différence des rangs entre les attributs des individus :

$$\rho(X,Y) = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \dots\dots (2)$$

Où $d_i = x_i - y_i$.

Pour pouvoir utiliser le résultat de ces deux mesures pour trouver le chemin Hamiltonien, nous avons appliqué une normalisation aux valeurs obtenues :

$$\frac{|1 - Résultat\ de\ mesure|}{2} \dots\dots (3)$$

Le résultat de cette normalisation inverse l'interprétation des mesures de corrélation de sorte que les données les plus proches auront la corrélation la plus petite (principe de TSP).

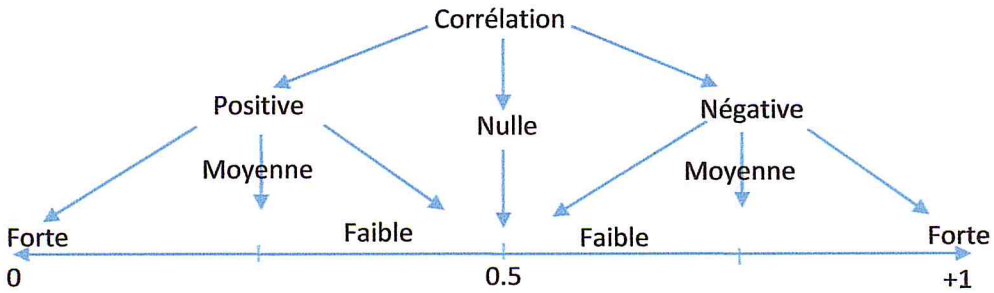


Figure III-10 : Interprétation de la corrélation après normalisation

- La distance de Minkowski

La distance de Minkowski est une mesure sur l'espace euclidien qui peut être considéré comme une généralisation à la fois la distance euclidienne (lorsque p=2) et la distance de Manhattan (lorsque p=1).

La distance de Minkowski à l'ordre p entre deux objets :

$$X = (x_1, x_2, \dots, x_n)\ et\ Y = (y_1, y_2, \dots, y_n)$$

Est définit par :

$$d(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \quad \ldots\ldots\ldots (4)$$

L'idée générale est que l'évaluation de la ressemblance entre deux objets peut être représentée d'un point de vue spatial. Les deux objets apparaissent alors comme des points dans l'espace ; et la similarité entre les deux objets est représentée par la distance entre ces deux points. Si deux sont rapprochés, cela voudra dire que les deux objets sont assez similaires [67].

- L'entropie

Les mesures citées précédemment (cf. & III.2.5.d) calculent la similarité des dimensions par paires et ne prennent pas en considération les groupes de données (clusters) et pour cela nous utilisant la mesure d'entropie proposée par K.Ameur [30].

L'entropie relative est une mesure de la distance entre les deux distributions de probabilité. Toutes ces quantités sont étroitement liées et partagent un certain nombre de propriétés simples.

La mesure suivante est proposée pour évaluer les dimensions, elle vise à ordonner les dimensions en séquentiel. Nous prenons en compte la répartition des clusters dans chaque valeur de dimensions X et Y. Le but de cette mesure est de réorganiser les dimensions selon le degré de séparation des clusters dans la façon dont l'utilisateur peut détecter le comportement semblable et différent des clusters.

$$D(X||Y) = \sum_{x\in X}\sum_{y\in Y} \frac{|c_{ix}|}{|x|} \sum_{i=0}^{|C|} \frac{|c_{ix} \cap c_{iy}|}{|xy|} \log\left(\frac{|c_{ix}|/|x|}{|c_{iy}|/|y|}\right) \quad \ldots\ldots\ldots (5)$$

Avec :

$\frac{|c_{ix} \cap c_{iy}|}{|xy|}$ C'est la valeur injectée dans la mesure. Cette mesure est définie dans R.

$D(X||Y) = 0$, c'est le cas où X et Y ont la même distribution de classes qui signifie X et Y ont le même comportement. Dans le cas où $D(X||Y) < 0$, X est mis avant Y. Cet ordonnancement des dimensions en fonction du degré de séparation des classes, nous pouvons dire X est plus général que Y. Sinon X sera mis après Y ($D(X||Y) > 0$).

L'entropie n'est applicable que pour les données classées dans le cas des données catégorielles à cause de sa qualité de solution lors de l'application sur des données continues (temps de calcul très long, résultat moins bon).

Après application d'une des mesures nous aurons la matrice de similarité contenant les distances entre les dimensions. Cette matrice se traduit par un graphe :

- Orienté (matrice non-symétrique) dans le cas de mesure d'entropie,
- Non-orienté (matrice symétrique) si nous appliquons les autres mesures,

	X	Y	U	V
X	d_{XX}	d_{Xy}	d_{XU}	d_{XV}
Y	d_{YX}	d_{YY}	d_{YU}	d_{YV}
U	d_{UX}	d_{UY}	d_{UU}	d_{UV}
V	d_{VX}	d_{VY}	d_{VU}	d_{VV}

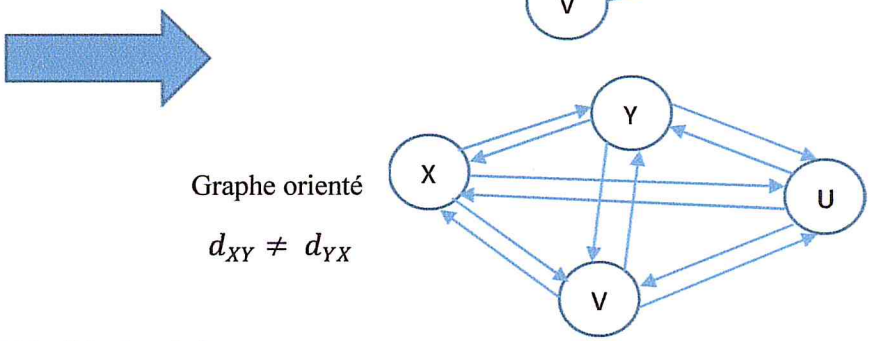


Figure III-11 : Résultat de la mesure

e) L'ordonnancement

Apres avoir eu la matrice de similarité (les poids (distance) entre les sommets (dimensions)), un algorithme de détection du plus court chemin peut être appliqué. Le plus court chemin trouvé définira l'ordre des dimensions.

➢ Ordonnancement par Branch & Bound (B&B).

Pour trouver la meilleure solution, nous avons implémenté l'algorithme de B&B. Le principe d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (les mauvaises explorations) et d'une technique pour orienter les différent choix.

Les principales fonctions de B&B sont la séparation et l'évaluation. Dans notre travail, nous avons adopté une stratégie de parcours du meilleur d'abord. Elle consiste à commencer la séparation de problème par le sommet du plus petit poids. Cette stratégie permet de trouver plus rapidement la meilleure solution pour empêcher l'exploration du plus grand nombre de chemins possibles.

L'évaluation fixe la borne supérieure que la meilleure solution peut avoir, tous les sommets ayant un poids égal ou supérieur à cette borne sont stérilisés (non explorables).

Algorithme :

Pour l'exploration des chemins, nous avons utilisé une structure TOUR qui a les paramètres suivant : le chemin, la profondeur, et le poids. Une liste de TOUR contiendra les tours à explorer.

1. Fixer un sommet de départ (non choisi à priori) et ajouter un chemin initial à la liste des tours. Si tous les sommets ont servi de sommets de départ, aller vers 8.
2. Si la liste n'est pas vide, tirer un tour de la liste après l'avoir ordonné par rapport aux poids des tours afin de commencer l'exploration par le meilleur d'abord. Si la liste est vide aller vers 1.
3. Si la profondeur de tour == taille du chemin, faire 4 sinon, faire 5
4. Si le poids du tour < la borne le poids du tour devient la nouvelle borne et le tour la meilleure solution.

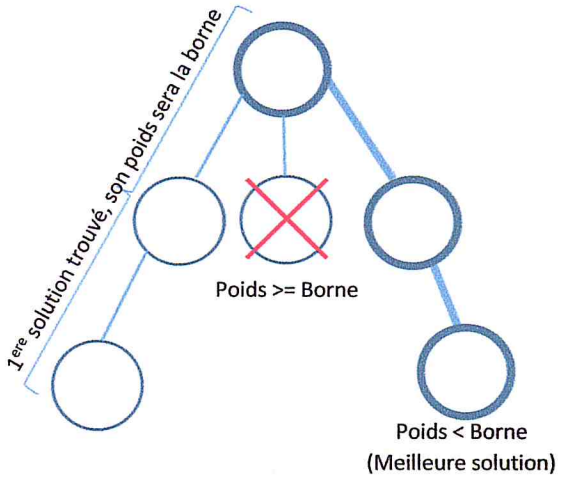


Figure III-12 : Construction de la solution en B&B

5. La profondeur du tour < taille de chemin veux dire que la solution n'est pas encore complètement construite alors nous poursuivons l'exploration. Si le poids du ce tour >= borne le tour est stérilisé sinon, faire 6.
6. Descendre dans la profondeur du tour courant et ajouter tous les chemins générés par la séparation du tour à la liste des tours à explorer.
7. Aller vers 2.
8. Fin (retourner la meilleure solution)

La complexité de cet algorithme au pire des cas est $\theta(n!)$.

Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si une solution proche de l'optimale répond au besoin [56]. Alors, nous avons implémenté l'algorithme de colonie de fourmis.

- Algorithme de colonie de fourmis (ACO) :

C'est un algorithme d'intelligence collective (essaim). En effet, bien que les fourmis ayant individuellement des capacités cognitives limitées, sont capables collectivement de trouver le chemin le plus court entre une source de nourriture et leur nid. Les fourmis utilisent l'environnement comme support de communication.

Règles utilisées :

Les paramètres de l'ACO sont :

- α : influence des phéromones (favorise l'exploration),
- β : influence des sommets adjacents (favorise l'exploitation),
- ρ : taux d'évaporation,
- Q : taux d'ajout de phéromone,
- m : nombre de fourmis.

Après avoir effectué des tests, nous avons fixé le paramètres de l'ACO à :

- $\alpha = 1$,
- $\beta = 2$,
- $\rho = 0.1$,
- $Q = 2.0$,
- $m = 20$.

La règle de déplacement, appelée « règle aléatoire de transition proportionnelle », est écrite mathématiquement sous la forme suivante :

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}(t)^{\alpha} * \eta_{ij}^{\beta}}{\sum_{l \in j_i^k} \tau_{il}(t)^{\alpha} * \eta_{ij}^{\beta}} & Si\ j \in J_i^k \\ 0 & Si\ j \notin J_i^k \end{cases} \quad \ldots\ldots\ldots (1)$$

Où J_i^k est la liste des déplacements possibles pour une fourmi k lorsqu'elle se trouve sur une ville i, η_{ij} la visibilité, qui est égale à l'inverse de la distance de deux villes i et j, $\left(\frac{1}{d_{ij}}\right)$ et $\tau_{ij}(t)$ l'intensité de la piste à une itération donnée t. Les deux principaux paramètres

contrôlant l'algorithme sont α et β, qui contrôlent l'importance relative de l'intensité et de la visibilité d'une arête.

$\Delta\tau_{ij}$ est la quantité de phéromone déposé par une fourmi k sur chaque arête lors de son parcours :

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^{k}(t)} \; Si \; (i,j) \in T^{k}(t) \\ 0 \quad Si \; (i,j) \notin T^{k}(t) \end{cases} \ldots\ldots\ldots (2)$$

Le système d'évaporation évite que les fourmis soient bloquées dans un chemin qui n'est pas le meilleur. À la fin de chaque itération de l'algorithme, les phéromones déposées aux itérations précédentes par les fourmis s'évaporent de $\rho\Delta\tau_{ij}^{k}$ ……..(3).

À la fin de l'itération, on a la somme des phéromones qui ne se sont pas évaporées et de celles qui viennent d'être déposées :

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\,\tau_{ij}^{k}(t) \ldots\ldots\ldots (4)$$

Algorithme :

1. Les m fourmis parcourent chacune un chemin de façon aléatoire et dépose les phéromones.
2. Tant que le nombre d'itération n'est pas atteint, faire 3 sinon, faire 10.
3. Une fourmi se met sur un sommet de départ (exploite le taux de phéromone pour chercher le plus court chemin).
4. Tant que la fourmi n'a pas parcouru tous les sommets, faire 5 sinon, faire 8.
5. La probabilité de déplacement est calculée pour chaque sommet suivant la formule 1.
6. L'arête ayant le plus grand taux de phéromone est choisie.
7. Mettre à jour le taux de phéromone :
    - Augmenter le taux de phéromone sur les arêtes prises par la fourmi (2).
    - Diminuer le taux de phéromone par l'effet d'évaporation (formule 3).
    - Le taux de phéromone final est calculé par la formule 4.
8. Calculer la longueur du chemin trouvé le retourner.
9. Si le chemin trouvé est meilleur que le précédent, il devient le meilleur.
10. Retourner le chemin dont la longueur est la plus courte.

Figure III-13 : Une itération de l'ACO

La complexité de l'ACO est $O(NI * n^2 * m)$, NI étant le nombre d'itération [68].

- Algorithme du plus proche voisin (NNF) :

Le NNF est un algorithme constructeur, il bâti la solution étape par étape. À chaque fois qu'il est sur un sommet, il cherche le plus proche voisin suivant et ainsi de suite jusqu'à la construction du chemin complet (solution).

Sa complexité est $\theta(n^3)$, définit par les trois boucles que nous avons utilisées pour le parcours des sommets.

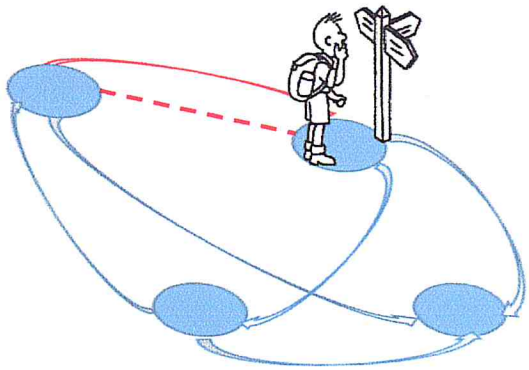


Figure III-14 : Construction de chemin en NNF

- ✓ La première boucle permet de changer le sommet de départ.
- ✓ La deuxième, de construire la solution.
- ✓ Et la boucle final permet de trouver le plus court chemin à chaque fois que nous nous trouvons sur un sommet.

III.2.6 Interaction

Pour faciliter l'exploitation des représentations de données, nous avons intégré des interactions à utiliser par le visionneur. Ces interactions sont nécessaires dans un outil d'analyse visuelle pour qu'il soit fiable. Ça intègre aussi l'expertise humaine dans le processus :

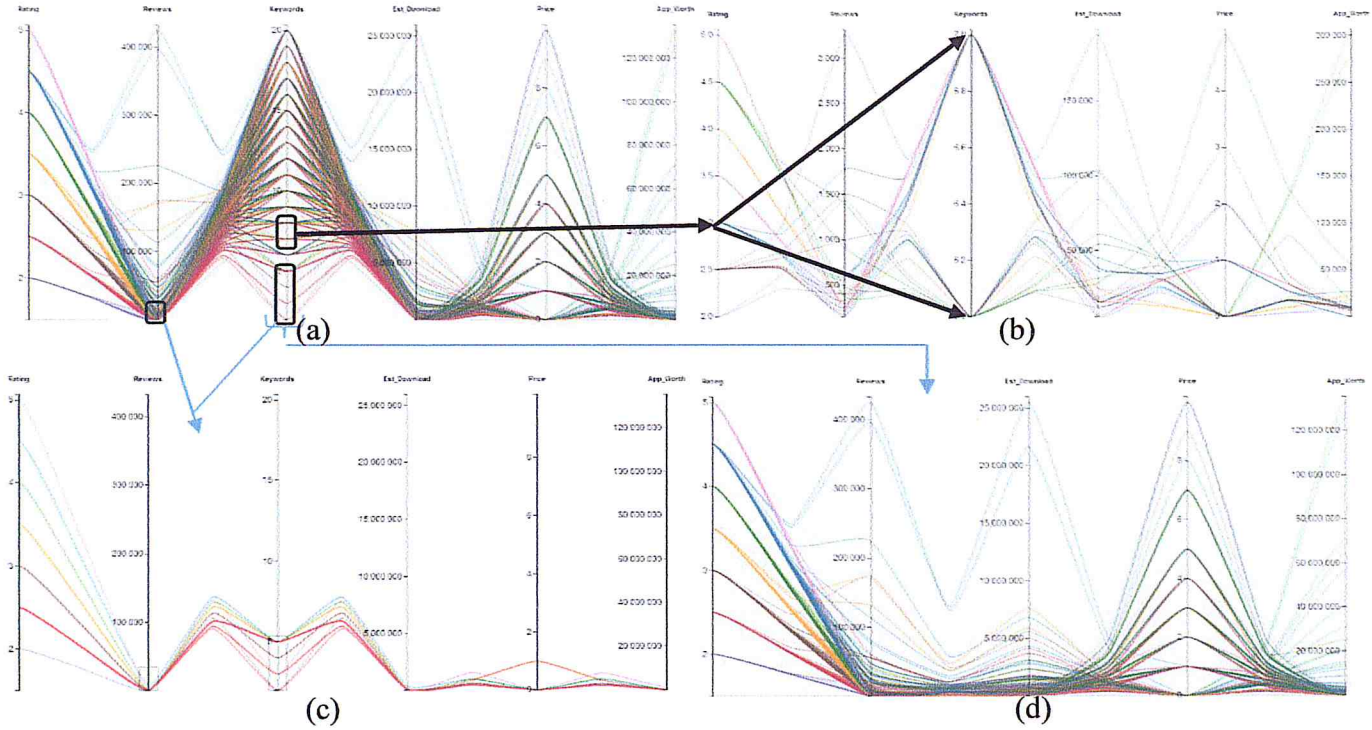


Figure III-15 : Résultat des interactions

- Filtrer (figure III-15-d)

Le filtrage permet aux utilisateurs d'enlever des graphes les dimensions (axes) non nécessaires à l'analyse, des dimensions qui n'ont pas une grande importance.

- Brushing (figure III-15-c)

Dans un affichage chargé de données, l'utilisateur peut visionner un sous ensemble d'éléments. Après avoir sélectionné ces derniers, les autres éléments sont cachés et il peut analyser l'ensemble des données choisi sur les différentes dimensions.

- Zoom (figure III-165-b)

Après avoir sélectionné un ensemble d'éléments (brushing), l'utilisateur demande plus de détails sur les données de l'ensemble. Un nouveau graphe est généré avec la valeur minimum et maximum de l'ensemble sélectionné comme bornes des dimensions sur ParCoords et les autres valeurs seront réparties entre les deux, seul l'ensemble sélectionné sera affiché sur tous l'écran (zoomant les éléments de l'ensemble).

- Réordonner manuellement les dimensions

L'utilisateur en tant qu'expert peut ordonner les dimensions manuellement d'après son point de vue. L'ordre choisi peut l'aider à mieux comprendre les données représentées dans le graphe et extraire les connaissances.

III.3 Conclusion

Après avoir introduit l'analyse visuelle des données et détailler la technique de ParCoords, nous avons vu les différentes méthodes d'ordonnancement des dimensions pour améliorer la représentation des données.

Au cours de ce chapitre nous avons intégré l'amélioration de l'analyse visuelle dans le processus proposé par D.Keim en utilisant les méthodes d'ordonnancement. Cette amélioration répond à notre problématique, elle aide l'analyste dans sa tâche de compréhension des informations et extraction de connaissances. L'analyse visuelle ne serait pas complète sans l'interaction humaine, alors nous avons impliqué l'homme dans le processus Afin d'exploiter ses compétences dans le bon déroulement de la tâche d'analyse.

Après avoir énuméré les étapes de notre travail, nous passons dans le prochain chapitre à l'implémentation. L'objectif de travail sera concrétisé dans un outil d'analyse visuelle doté d'une intelligence artificielle offrant une meilleure représentation des données. L'implémentation sera suivie d'une série de tests afin d'évaluer le travail réalisé ainsi que les résultats obtenus.

Chapitre

IV. Implémentation, tests et résultats

IV.1 Introduction

La dernière partie de travail est la mise en œuvre d'un outil de visualisation implémentant les résultats de la recherche et les méthodes choisies et montrant l'impact de notre approche. Pour l'environnement de développement, nous avons choisi le web. Le web est le moyen le plus rapide pour que le travail soit vu et critiqué en atteignant un public mondial. Travailler avec les technologies standards Web signifie que votre travail peut être vu et vécu par toute personne utilisant un navigateur Web récent, quel que soit le système d'exploitation (Windows, Mac, Linux) et le type d'appareil (ordinateur portable, ordinateur de bureau, Smartphone, tablette). En évitant les logiciels propriétaires et les plug-ins, vous pouvez vous assurer que vos projets sont accessibles sur le plus large éventail possible d'appareils.

En plus de cet atout, le web offre des fonctionnalités et des bibliothèques graphiques qui permettent de faciliter le travail et atteindre un bon résultat.

IV.2 Implémentation

IV.2.1 Langages et outils de développement

JavaScript (JS) est le langage de programmation du Web. La grande majorité des sites web modernes utilisent JavaScript, et tous les navigateurs web modernes sur PC, consoles de jeux, les tablettes et les téléphones intelligents, incluent un interpréteur JavaScript, ce qui fait de JavaScript le langage de programmation le plus répondu dans l'histoire. JavaScript est partie de la triade des technologies que tous les développeurs Web doivent apprendre : HTML pour spécifier le contenu des pages Web, CSS pour spécifier la présentation des pages Web, et JavaScript pour spécifier le comportement de pages Web.

Plusieurs bibliothèques de visualisation ont été développées en JS. Dans notre implémentation nous avons utilisé D3.js pour la facilité qu'elle offre dans la mise en œuvre des coordonnées parallèles.

D3 est une bibliothèque JavaScript qui permet de manipuler des documents basés sur des données. D3 nous aide à donner vie aux données en utilisant le HTML, SVG et CSS.

Cette bibliothèque est écrite en JavaScript et utilise un style fonctionnel qui nous a permis de réutiliser le code et ajouter des fonctions spécifiques au contenu de notre travail. En d'autres termes, il permet de construire le *Framework* de visualisation de données d'une manière flexible. Elle peut être ajoutée au front-end (côté client) de notre application web pour que l'utilisateur interagisse avec l'application. Ou notre back-end (côté serveur) pour ne générer que les données nécessaires [69].

IV.2.2 Architecture générale

Afin d'organiser notre travail, nous avons décomposé notre programme en modules, chaque groupe de modules spécifié pour une étape de processus de visualisation.

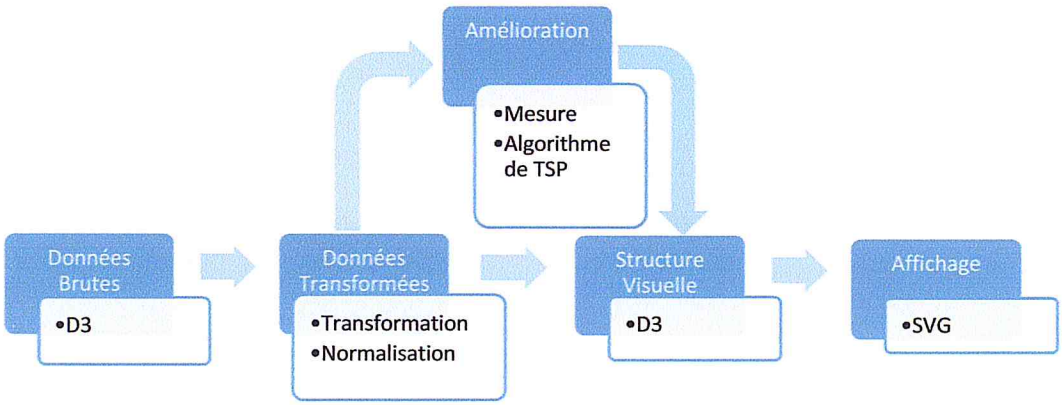


Figure IV-1 : Les modules utilisés à chaque étape de processus de visualisation

Etape	Module	Description
Données brutes	D3	D3 (bibliothèque) comporte une partie d'intégration des données de différents formats et détecte la ligne contenant les dimensions.
Données transformées	Transformation	Module implémenté pour la transformation des données qualitatives.
	Normalisation	Le module de normalisation des données
Structure Visuelle	D3	D3 construit la structure visuelle de ParCoords
Affichage	SVG	Le SVG pour le dessin et l'affichage des structures à l'écran.
Amélioration	Mesure	Ce module contient les différentes mesures implémentées.
	Algorithme de TSP	Module des algorithmes d'ordonnancements

Tableau IV-1 : Description des modules

IV.2.3 Présentation de ClusterViz

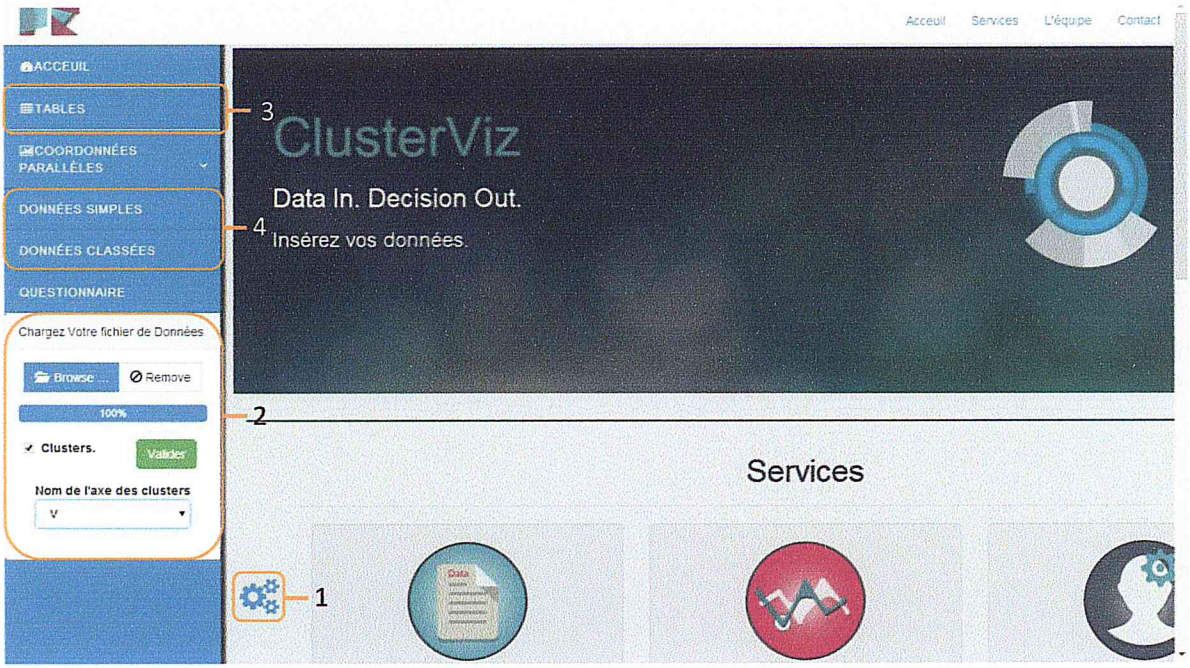


Figure IV-2 : Accueil et menu de ClusterViz

Le bouton de menu (1) donne l'accès aux fonctionnalités de ClusterViz. Commençant par l'importation des données dans la partie de recharge de données(2). Après l'importation, l'utilisateur peut choisir la dimension qui représentera les classes de données (clusters).

La table de données sera affichée à l'utilisateur où il peut voir et modifier le contenu (3).

Ensuite, il entame l'étape de la visualisation. Sur la partie (4) du menu, l'utilisateur a le choix entre le ParCoords efficace pour les données continues ou ParSets meilleur avec les données catégorielles.

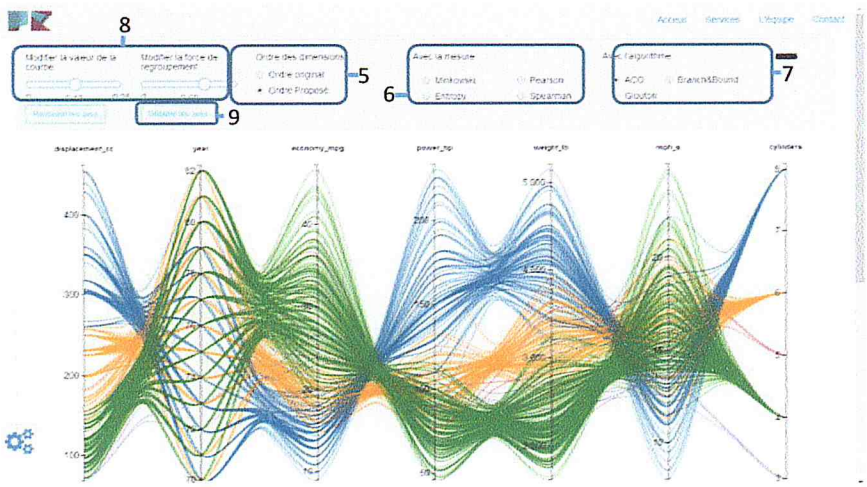


Figure IV-3 : La représentation des données avec ClusterViz

Le bouton de choix d'ordre (5) ouvre l'angle pour définir la mesure de calcul de la similarité (6), et l'algorithme d'ordonnancement(7).

L'utilisateur interagit directement avec le graphe pour effectuer les différentes opérations (filtre, ordonnancement manuel, brushing). Lors de l'utilisation du brushing, les données sont zoomées (8). La partie (9) définit la force de regroupement et la valeur des courbes.

Dans le cas où les données sont classées, l'utilisateur aura un menu pour choisir la représentation par MinMax.

IV.3 Tests et résultats

Le principal critère dans les tests est le bon ordonnancement des dimensions pour mieux interpréter les représentations. En plus de cela, les tests porteront sur le temps d'exécution des algorithmes d'ordonnancement.

A cause du temps de calcul de la mesure d'entropie et de son efficacité avec les données continues, nous séparons les tests par rapport aux données utilisées.

Les tests sont effectuées sur un ordinateur ayant un CPU Core deux duo 2.8 Ghz et 2 GO de Ram.

IV.3.1 Données utilisées

a) Auto MPH

Pour tester les performances de ClusterViz nous avons utilisé un *Benshmark* de *Dataming* nommé *Auto MPH* [70].

Les données concernent la consommation de carburant en miles par gallon, qui est prédite en fonction de 3 attributs discrets et 5 continus. Le nombre d'éléments est de 398 et 9 dimensions (Quinlan, 1993).

Dimension	Type	**Description**
mph	continue	vitesse
cylinders	discret	nombre de cylindre
displacement	continue	la vitesse de mouvement des soupapes
horsepower	continue	nombre de chevaux
weight	continue	le poids
acceleration	continue	acceleration
model year	discret	l'année mise en circulation
origin	discret	le pays d'origine
name	discret (unique pour chaque élément)	nom de la voiture

Tableau IV-2 : La description des dimensions d'auto mph

b) Données proposées

Ces données sont utilisées pour montrer l'impact de la mesure d'entropie sur la représentation des données [30].

V	X	Y	W	U	S	T	Z	Class
2	0	2	2	1	0	2	0	2
2	0	2	2	1	0	1	3	2
1	0	1	2	2	3	3	0	1
3	0	1	2	3	3	1	3	3
2	1	2	1	1	0	1	0	2
1	1	1	1	2	2	3	3	1
1	1	1	1	2	2	3	0	1
3	1	2	1	3	2	2	3	3

Tableau IV-3 : Table des données

c) Iris

L'ensemble de données contient 3 classes de 50 cas chacun, où chaque classe se réfère à un type de plante de l'iris.

Dimension	Type	Description
Sepale lenght (cm)	discret	Longueur du sépale
Sepale width (cm)	discret	Largeur du sépale
Petale lenght (cm)	discret	Longueur du pétale
Sepale width (cm)	discret	Largeur du pétale

Tableau IV-4 : Description des dimensions d'Iris

IV.3.2 Application des tests

a) Auto mph

La représentation des données d'auto mph sur le ParCoords donnera le graphe suivant :

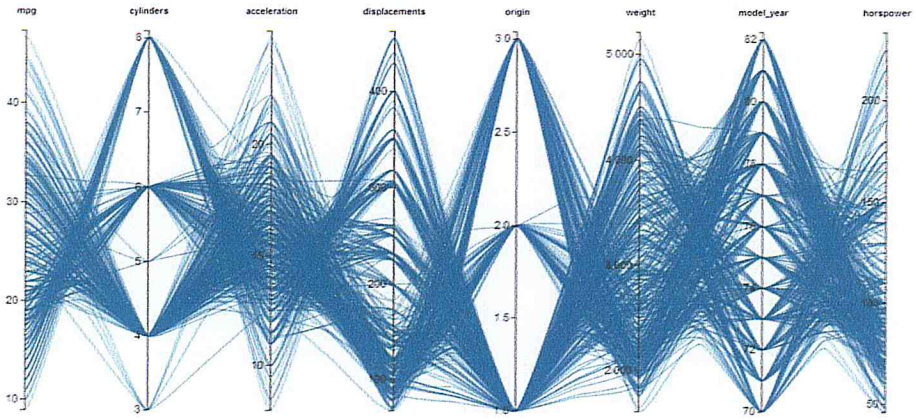


Figure IV-4 : La représentation des données (auto mph) sur ParCoords

Pour améliorer la visualisation, les données sont classées par cylindre avec l'utilisation des courbes.

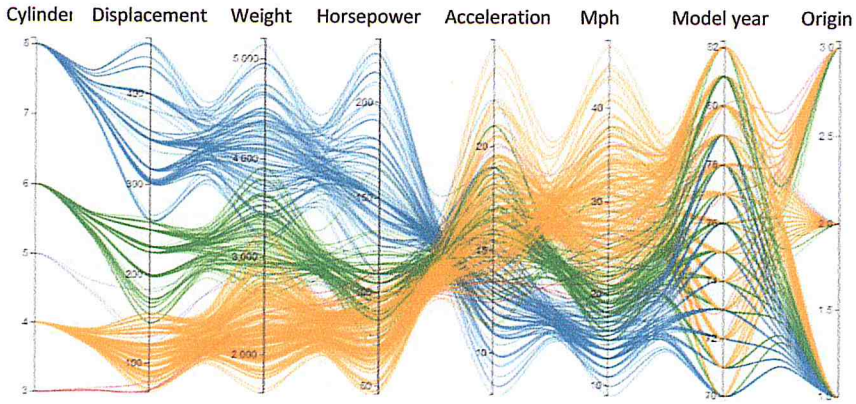
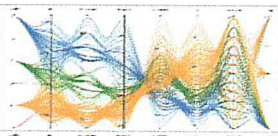
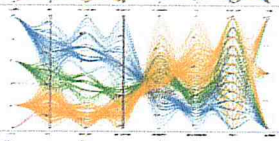
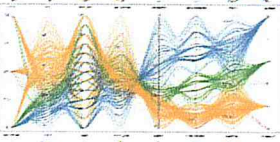
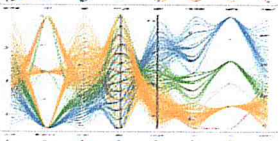
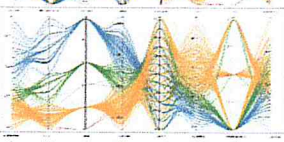
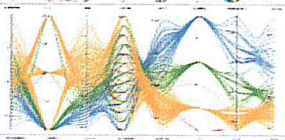
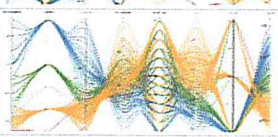
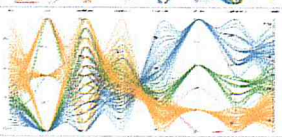


Figure IV-5 : Meilleur ordre des dimensions (auto mph)

La figure IV-5 représente le meilleur ordre de dimensions pour ces données, l'évaluation des différents ordres lors des tests sera effectuée par rapport cette dernière.

Nous notons les résultats des tests dans le tableau suivant.

Mesure	Algorithme d'ordonnancement	Temps d'ordonnancement	Ordre	Evaluation
Minkowski P=2	ACO	34 ms		Tres bon
	NNF	1 ms		Tres bon
	B&B	115 ms		Bon
Pearson	ACO	33 ms		Bon
	NNF	1 ms		Bon
	B&B	34 ms		Bon
Spearman	ACO	40 ms		Moin bon
	NNF	1 ms		bon

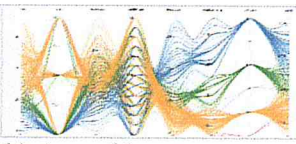
	B&B	58 ms		bon

Tableau IV-5 : Résultats des tests (Auto mph)

➢ Interprétation

Dès que nous regardons la figure IV-5, nous pouvons distinguer les relations entre les données, leur comportement commun ainsi que leur différence. Ce qui nous permet de déduire dans le cadre de la consommation des voitures, que les plus lourdes voitures possédant un puissant moteur ont la plus grande consommation de carburant (sachant que les grandes cylindrés consomment beaucoup de carburant). Nous extrayons d'autre information qui stipulent que malgré leur puissant moteur, les voitures lourdes sont moins rapides.

Nous interprétons par la suite le résultat de chaque mesure. Le sens de la lecture du graphe n'est pas important car la matrice de similarité en appliquant ces mesures est symétrique.

Après avoir appliqué les différentes mesures et algorithmes sur le graphe (figure IV-4), les résultats étaient plutôt bons, et ordonnent les dimensions étaient ordonnées de façon à montrer les comportements communs des données et les différences, et la compréhension et l'interprétation des informations à l'utilisateur sont devenues plus faciles.

Minkowski calcule la distance entre les données ce qui a permis de montrer le comportement commun des données (petite distance) et les données non similaires (grande distance).

Pearson calcule la corrélation entres les données ce qui ordonne les données similaire ayant une corrélation positive, les données similaires ayant une corrélation négative et entre les deux, la transition (les données non similaires).

Quant à Spearman, Il donne un résultat similaire à Pearson dans ce test, mais ce n'est pas toujours le cas car contrairement à Pearson, Spearman utilise le rang des valeurs.

b) Données proposées

Pour le test de la mesure d'entropie nous avons utilisé les données présentées précédemment. Leur meilleur ordre est le : X – W – Z – Y – T – S – U – V.

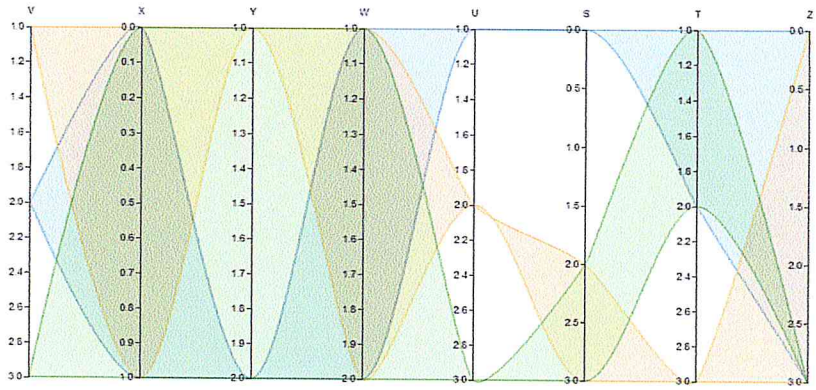


Figure IV-6 : Représentation des données

Afin de mieux visualiser les données nous utilisons la représentation par MinMax.

Nous choisissons la distance de Minkowski comme mesure afin de voir l'impact de l'entropie sur l'ordonnancement des dimensions.

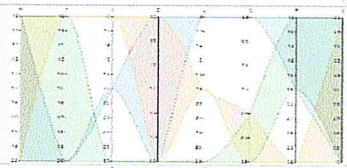
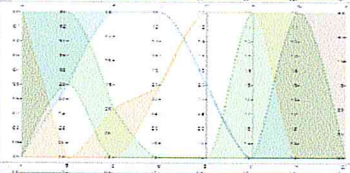
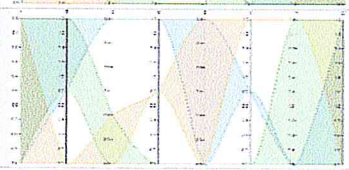
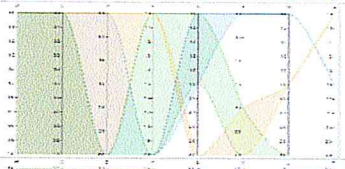
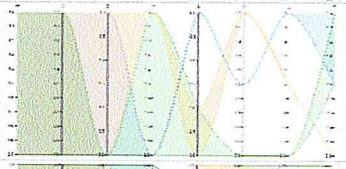
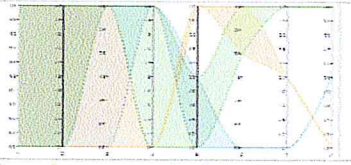
Mesure	Algorithme d'ordonnancement	Temps d'ordonnancement	Ordre	Evaluation
Minkowski P=2	ACO	55 ms		Movais
	NNF	1 ms		Movais
	B&B	5900 ms		Movais
Entropie	ACO	52 ms		Tres bon
	NNF	1 ms		Moins Bon
	B&B	44 ms		Tres bon

Tableau IV-6 : Résultats des tests (données proposées)

➢ Interprétation

L'ordre obtenu en appliquant la distance de Minkowski est mauvais et ne répond pas au besoin (généralisation vers spécification). Cette mesure ne prend pas en considération les clusters ce qui donne un mauvais résultat lors de l'application sur ces données.

L'application de l'entropie a donné de meilleurs résultats avec l'ACO et B&B mais pas le NNF.

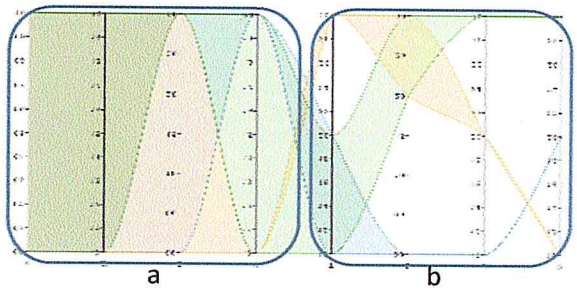


Figure IV-7 : Meilleur ordre des données de test

Le meilleur ordre obtenu montre clairement l'effet de cet ordre dans la visualisation des clusters et comment réorganisé les groupes de la façon à ce que l'utilisateur peut détecter le comportement semblable et différent des clusters. La figure IV-7 montre l'efficacité de cette mesure pour réorganiser les dimensions en fonction du degré de séparation des groupes de la dimension (la plus générale à la plus spécifique). La partie b de la figure IV-7 détermine les dimensions qui caractérisent les clusters [30].

c) Iris

Ces *Benshmark* nous permettrons de tester la mesure d'entropie sur des données concrètes et interpréter le résultat.

L'ordonnancement des dimensions d'Iris nous donne la représentation suivante :

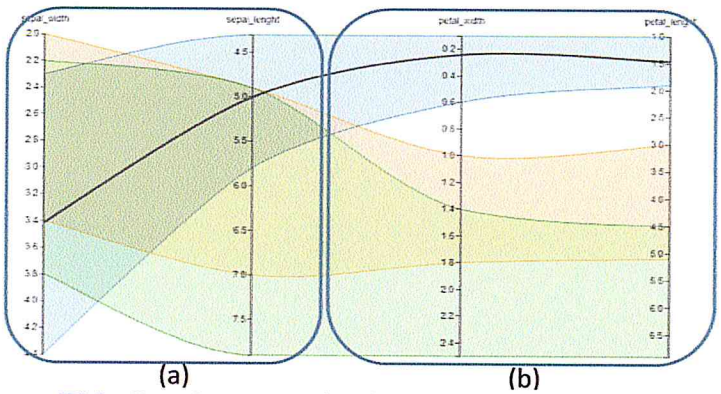


Figure IV-8 : Représentation des données d'Iris sous le meilleur ordre

Interprétation :

Après ordonnancement nous distinguons les deux parties lors de l'application d'entropie:

- La partie générale qui ne représente pas des axes spécifique au class d'Iris (figure IV-8-a).
- La partie spécifique qui désigne les class (figure IV-8-b).

De graphe, nous observons que la taille des sépales des plantes représentées est commune entre eux, et que c'est la taille des pétales qui change.

IV.4 Conclusion

Dans ce chapitre, nous avons commencé par montrer l'aspect programmation de ClusterViz et le découpage de l'implémentation en modules pour chaque étape du processus de visualisation pour une meilleure organisation. La mise en œuvre de L'application est suivie par des tests d'évaluation. Cela nous a permis de spécifier pour chaque type de données, les mesures qui lui conviennent. À cause de temps du calcul de l'entropie avec les données continues, nous étions obligés de deviser les tests en deux parties. Toutefois, les tests ont montré l'impact de l'ordonnancement des dimensions sur l'interprétation des graphes et l'extraction des connaissances. Les résultats obtenus étaient très satisfaisants et ont démontré la fiabilité des algorithmes implémentés pour l'amélioration de la représentation des données.

Conclusion générale

Conclusion générale

Le sujet d'étude de ce mémoire concernait l'amélioration de l'analyse visuelle des données multidimensionnelles représentées avec les coordonnées parallèles par ordonnancement des dimensions. La difficulté de représenter les données sur les coordonnées parallèles lorsque ces dernières sont de grande taille, exige des méthodes de réduction d'encombrement dans l'affichage causé par les grandes masses de données. Plusieurs méthodes ont été proposées afin d'améliorer la visualisation des données pour optimiser l'interprétation et l'extraction des connaissances dont la visualisation des clusters au lieu des données, la réduction des dimensions, etc. Notre approche consiste à ordonner les axes représentant les dimensions des données sur les coordonnées parallèles dans le but de réduire l'encombrement qui est le facteur major responsable de rendre l'interprétation des graphes difficile, et de mieux apercevoir les relations entre les données où nous pouvons voir le comportement commun et différent des données. En quête de proposer le meilleur ordre pour une interprétation facile, nous avons lié le problème d'ordonnancement des dimensions au problème du voyageur de commerce qui consiste à trouver un chemin Hamiltonien.

Nous avons débuté le mémoire en présentant le domaine de l'analyse visuelle des données et compris son importance lorsqu'il s'agit de comprendre et utiliser un grand nombre de données. Le deuxième pas consistait à étudier le problème de voyageur de commerce afin de trouver une solution au problème d'ordonnancement. Après avoir déterminer les solutions possibles au voyageur de commerce et calculer la matrice de similarité entre les dimensions, nous les avons couplé au problème d'ordonnancement des dimensions. Pour terminer, nous avons implémenté notre approche dans un outil d'analyse visuelle des données qui propose un meilleur ordre de dimensions aidant l'utilisateur dans le processus d'extraction des connaissances. Afin de voir l'impact de nos travaux sur l'interprétation des données nous avons effectué des tests sur différents ensembles de données.

Pour conclure, nous dirons que les objectifs de départ ont été atteints et la problématique a été résolue par une proposition qui a donné des résultats encourageants mais qui peuvent, toutefois, faire l'objet d'améliorations. Ainsi, nos perspectives sont les suivantes :

Nous travaillons sur la normalisation de la mesure d'entropie pour qu'elle soit applicable sur des données continues en but de réduire les croisements entre les données.

Concernant la partie de logiciel, nous proposons d'ajouter le *Clustering* et d'autres techniques de représentation de données commençant par les plus familières (les plus connues et utilisées) afin de publier ClusterViz comme un logiciel d'analyse visuelle des données.

Cette étude nous a permis d'une part, de découvrir le domaine de la visualisation de données comme étant un secteur en plein évolution et de plus en plus intégré en raison de l'immensité des données stockées dans le monde, et de connaitre la technique des coordonnées parallèles afin de l'améliorer pour une meilleure utilisation. Elle nous a aussi donnée l'occasion de s'approfondir dans un des problèmes d'optimisation vue durant notre cursus universitaire et lui implémenter des solutions.

Bibliographie

[1] D. A. Keim, «Information Visualization and Visual Data Mining,» *IEEE,* vol. 7, n° %11, pp. 100-107, 2002.

[2] A. Inselberg, Parallel coordinates: Visual multidimensional and its applications, New York: Springer, 2009, p. 361–378.

[3] J. W. Tukey, Exploratory Data Analysis, Boston: Addison-Wesley, 1977.

[4] D. Keim, J. Kohlhammer, G. Ellis et F. Mansmann, Mastering the information age: Solving problems with visual analytics, Goslar: Druckhaus "Thomas Müntzer" GmbH, Bad Langensalza, 2010.

[5] B. Shneiderman, «The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations,» *IEEE,* vol. 96, n° %15, pp. 336-343, 1996.

[6] S. Vidal, «Visualisation de l'information: Un panorama d'outils et de méthodes,» Centre national de la recherche scientifique, Paris, 2006.

[7] W. Peng, *Clutter-Based Dimension Reordering in Multi-Dimensional Data Visualization,* Worcester, January 2005.

[8] J.-D. Fekete, «La visualisation analytique, pour comprendre des données complexes,» 2011 05 05. [En ligne]. Available: https://interstices.info/vismaster. [Accès le 10 05 2014].

[9] [En ligne]. Available: http://mashable.com/category/data-visualization/ . [Accès le 17 05 2014].

[10] R. Mazza, «Intdrocution to information vizualisation,» University of Lugano, Lugano, 2004.

[11] Z. Bin et C. Hsinchun, «Information Visualization,» *Annual Review of Information Science and Technology* , vol. 40, pp. 139-177, 2004.

[12] D. A. Keim, J. Schneidewind, H. Ziegler et F. Mansmann, «Challenges in visual data analysis,» chez *International Conference on Information Visualisation (IV)*, London, 2006.

[13] G. Jaeschke, P. Gupta et M. Hemmje, Modelling Interactive, Three-Dimensional information visualizations, Berlin: Springer, 2005.

[14] M. Dias Maria, J. K. Yamaguchi, E. Rabelo et C. Franco, Visualization Techniques: Which is the Most Appropriate in the Process of Knowledge Discovery in Data Base, Paraná: State University of Maringá, 2012.

[15] G. Georges, T. Marjan et C. Urška, «High-Dimensional Visualizations,» chez *KDD*, 2002.

[16] K. T. McDonnell et K. Mueller, «Illustrative Parallel Coordinates,» *IEEE,* vol. 27, n° %13, 2008.

[17] GGobi , «GGobi: Out of sight, out of mind,» GGobi, 20 02 2006. [En ligne]. Available: http://www.ggobi.org/. [Accès le 20 05 2014].

[18] The ROOT team, «ROOT Project Founders,» 1995. [En ligne]. Available: http://root.cern.ch. [Accès le 2014 05 20].

[19] D.Brodbeck; L.Girardin, «Interactive data visualization,» macrofocus, 2000. [En ligne]. Available: www.high-d.com/. [Accès le 20 05 2014].

[20] TIBCO Spotfire, «Spotfire,» TIBCO, 18 09 2013. [En ligne]. Available: www.stn.spotfire.com. [Accès le 20 05 2014].

[21] C. Chih-Chung et L. Chih-Jen, «orange,» 2000. [En ligne]. Available: www.orange.biolab.si/. [Accès le 20 05 2014].

[22] «XmdvTool Home page,» Xmdv, 1997. [En ligne]. Available: www.davis.wpi.edu/xmdv/. [Accès le 20 05 2014].

[23] Y. Xiang, D. Fuhry, R. Jin, Y. Zhao et K. Huang, «Visualizing Clusters in Parallel Coordinates for Visual Knowledge Discovery,» *PAKDD,* pp. 505-514, 2012.

[24] W. Matthew, G. George et K. Daniel, Interactive Data visualization Book: Foundation, Technique, Application, 2010.

[25] E. J. Wegman, «Hyperdimensional data analysis using parallel coordinates,» *Journal of the American Statistical Association,* vol. 85, n° %1411, p. 664–675, 1990.

[26] Y. H. Fua, M. O. Ward et E. A. Rundensteiner, «Hierarchical Parallel Coordinates for Exploration of Large Datasets».

[27] Z. Hong, Y. Xiaoru, Q. Huamin, C. Weiwei et C. Baoquan, «Visual Clustering in Parallel Coordinates,» *IEEE,* vol. 27, n° %13, 2008.

[28] «Tetherless World Constellation,» 14 11 2012. [En ligne]. Available: www.twcmaxcurran.blogspot.com/. [Accès le 30 05 2014].

[29] Y. Luo, D. Weiskopf, H. Zhang et A. E. Kirkpatrick, «Cluster Visualization in Parallel Coordinates Using Curve Bundles,» *IEEE,* pp. 1-11, 2008.

[30] K. Ameur, N. Benblidia et S. Oukid-Khouas, «Enhanced Visual Clusters Analysis by Dimensions Reordering in Parallel Coordinates,» *IEEE,* pp. 1-4, 2013.

[31] B. Roerdink et J. Ferdosi, Visualizing High-Dimensional Structures by Dimension Ordering and Filtering using Subspace Analysis, Bergen, 2011.

[32] T. Boogaerts, L.-C. Tranchevent, A. P. Georgios, J. Aerts et J. Vandewalle, «Visualizing High Dimensional Datasets Using Parallel Coordinates: Application to Gene Prioritization».

[33] L. N. Biggs, J. Lloyd, E. Keith et W. Robin, Graph Theory 1736-1936, Oxford: Clarendon Press, 1986.

[34] E. L. Lawler, J. K. Lenstra, A. H. G. Shmoys, K. Rinnooy et B. D, The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, 1985.

[35] G. B. Dantzig, R. Fulkerson et S. M. Johnson, Solution of a large-scale traveling salesman problem, 1954.

[36] P. Cristophe et S. Thomas, «Problème du voyageur de commerce et algorithmes génétiques,» 2001.

[37] T. Barthélemy, N. Coindet et C. Fall, «Résolution du problème du voyageur de commerce asymétrique par séparation et évaluation de sa relaxation combinatoire en problème d'affectation,» Université de Nante, Nante, 2011.

[38] M. Rajesh, P. S. Surya et L. M. Murari, «Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches,» InTech, New Delhi, 2010.

[39] M. Grötschel, M. Jünger et G. Reinelt, «Optimal Control of Plotting and Drilling Machines: A Case Study,» *Mathematical Methods of Operations Research,* January 1991.

[40] R. D. Plante, T. J. Lowe et R. Chandrasekaran, «The Product Matrix Traveling Salesman Problem: An Application and Solution Heuristics,» *Operations Research,,* 1987.

[41] R. Bland et D. Shallcross, «Large traveling salesman problem arising from experiments in X-ray crystallography: a preliminary report on computation,» *Operations Research Letters,* 1989.

[42] W. Dreissig et W. Uebaeh, «Personal communication,» 1990.

[43] J. K. Lenstra, A. H. G. Problem et K. Rinnooy, «Some Simple Applications of the Travelling Salesman,» *BW Stichting Mathematisch Centrum,* vol. 38, n° %174.

[44] H. A. Saleh et R. Chelouah, «The design of the global navigation satellite system surveying networks using genetic algorithms,» *Engineering Applications of Artificial Intelligence,* 2004.

[45] «An automatic method for solving discrete programming problems,» *Econometrica,* vol. 28 , p. 497–520, 1960.

[46] P. Fouilhoux, *Programmation mathématique Discrète et Modèles Linéaires,* Paris: Université Pierre et Marie Curie, 2013.

[47] A. Schrijver, «Theory of linear and integer programming,» *Wiley and Sons,* 1986.

[48] S. Douiri, S. Elbernoussi et H. Lakhbab, *Cours des Méthodes de Résolution Exactes Heuristiques et Métaheuristiques,* Rabat: Université Mohammed V.

[49] F. Glover, G. A. Kochenberger et B. Alidaee, «Adaptive memory tabu search for binary quadratic programs,» *Management Science,* vol. 44, p. 336–345, 1998.

[50] P. Hansen, «The steepest ascent mildest descent heuristic for combinatorial programming,» présenté au Congress on Numerical Methods in Combinatorial Optimization, Capri, Italie, 1986.

[51] F. Laguna et M. Glover, Tabu search, Norwell MA: Kluwer Academic Publishers, 1997.

[52] V. Gardeux, *Conception d'heuristiques d'optimisation pour les problèmes de grande dimension,* Paris: UNIVERSITÉ DE PARIS-EST CRÉTEIL, 2011.

[53] S. Kirkpatrik et C. D. Gelatt, *Optimization by simulated annealing,* 1983.

[54] V. Cerný, «A thermodynamical approach to the traveling salesman problem an efficient simulation algorithm,» *Journal of Optimization Theory and Applications,* 1985.

[55] N. Metropolis, M. N. Rosenbluth et H. A, «Equation of state calculation by fast computing machines,» *Journal of Chemical Physics,* vol. 21, n° %16, 1953.

[56] J. K. Hao, P. Galinier et M. Habib, «Méthaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes,» *Revue d'Intelligence Artificielle,* pp. 1-28, 1999.

[57] S. B. Ismail, *Introduction à l'optimisation combinatoire,* 2012.

[58] T. B. Tadunfock et L. P. Fotso, «Heuristiques du problème du voyageur de commerce,» chez *CARI06*, Yaoundé, 2006.

[59] *Dévéloppement d'un algorithme de type voyageur de commerce généralisé pour un problème de trajet optimal dans une ville,* MONTRÉAL: UNIVERSITÉ DU QUÉBEC À MONTRÉAL, 2011.

[60] V. Berry, *Algorithmes stochastiques,* Master IC, 2009.

[61] J. L. Deneubourg, S. Aron, S. Goss et J. M. Pasteels, «The self-organizing exploratory pattern of the argentine ant,» *Journal of Insect Behavior,* vol. 3, n° %12, pp. 159-168, 1990.

[62] M. Dorigo, *Optimization, Learning and Natural Algorithms,* Milan: PhD thesis, Politecnico di Milano, 1992.

[63] S. DJELLAT, *Optimisation Par Colonie de Fourmies,* Oran: Université des Sciences et de la Technologie d'Oran, 2012.

[64] K. Robert, B. Fabian et H. Helwig, «Parallel Sets: Interactive Exploration and Visual Analysis of Categorical Data,» *IEEE,* vol. 12, n° %14, pp. 1-11, 2006.

[65] J. Adler, R, L'essentiel, Montreuil: Pearson Education, 2011.

[66] «Chapitre 6 : LA CORRELATION,» Université de Paris, [En ligne]. Available: http://grasland.script.univ-paris-diderot.fr/STAT98/stat98_6/stat98_6.htm. [Accès le 05 Juin 2014].

[67] J. P. Vandegeer, Some aspect of Minkowski distance, Leiden, Pays-Bas: Leiden university.

[68] M. Dorigo, V. Maniezzo et A. Colorni, «The Ant System: Optimization by a colony of cooperating agents,» *IEEE Transactions on Systems, Man, and Cybernetics.*

[69] D. Flanagan, JavaScript the definitive guide, Sebastopol, CA: O'Reilly, 2011.

[70] D. Aha et f. graduate, «The UCI Machine Learning Repository,» Center for Machine Learning and Intelligent Systems, 1987. [En ligne]. Available: www.archive.ics.uci.edu/ml/datasets.html. [Accès le 04 06 2014].

[71] [En ligne]. Available: http://bdtnp.lbl.gov/Fly-Net/content/bid/pcx/ParallelCoordinates/ParallelCoordinates.html#Properties3. [Accès le 25 05 2014].

[72] M. D. Ocagne, Coordonnées parallèles & axiales: méthode de transformation géométrique, Cambridge: Gauthier-Villars, 1885.