



Projet Calcule scientifique

1GI

Problème de voyageur de commerce

Réalisé par :

- AMCHAROD YASSINE
- AMGHAR SOUHAIL
- RHOUNIMI AISSAM
- ETANGI SALAH-EDDINE

Encadrant :

- Pr. Salem NAFIRI

Tables de matières :

Introduction	3
Présentation du problème de voyageur de commerce.....	4
I. Notion de base sur la théorie des graphes.....	6
II. Etude mathématique du problème	11
III. Implémentation numérique du problème.....	15
IV. Champs d'application.....	23
Conclusion.....	26
Bibliographie.....	27

Introduction :

Dans le cadre du projet final de l'élément de module "Calcul scientifique" nous avons choisi de traiter le problème de voyageur de commerce. L'énoncé du problème est si simple ; Le "problème du voyageur de commerce", ou TSP (pour Traveling Salesman Problem), est le suivant : « un représentant de commerce ayant n villes à visiter souhaite établir une tournée qui lui permette de passer exactement une fois par chaque ville et de revenir à son point de départ pour un moindre coût, c'est-à-dire en parcourant la plus petite distance possible ». Ce sujet a suscité notre intérêt d'abord pour sa popularité et pour la simplicité de son énoncé, mais surtout pour la multitude d'aspects que nous devions traiter pour la résolution de notre problème. Parmi lesquelles nous citons :

- Une initiation à l'étude des graphes.
- L'étude des algorithmes qui donnent la solution à ce problème.
- L'implémentation de ces algorithmes sous forme d'un programme informatique.
- La mise en place d'une interface graphique optimale et interactive pour le programme, permettant d'aider un commerçant à visiter plusieurs villes marocaines en traversant le plus court chemin.

Au départ, nous proposons une présentation du problème de voyageur de commerce, ensuite nous aborderons une étude mathématique de la solution de ce problème, ensuite nous présenterons un algorithme pour la résolution numérique du problème, et enfin nous allons citer quelques applications du problème de voyageur de commerce.

Présentation du problème :

- Le terme *problème du voyageur de commerce*, vient de la traduction de l'anglais américain *Traveling salesman problem*, qui est apparu dans les années 1930 ou 40, sans doute à l'université de Princeton où plusieurs chercheurs s'y intéressaient. C'est aussi à cette période que le problème est formulé indépendamment dans plusieurs communautés de chercheurs, notamment autour de mathématicien Karl Menger.
- Le voyageur de commerce est aujourd'hui l'un des problèmes algorithmiques les plus étudiés.
- Ce problème d'optimisation combinatoire appartient à la classe des problèmes NP-Complets (solution non polynomiale ; il n'y a pas une formule spécifique pour donner la réponse).
- Il existe deux grandes catégories de méthodes de résolution : les méthodes exactes et les méthodes approchées. Les méthodes exactes permettent d'obtenir une solution optimale à chaque fois, mais le temps de calcul peut être long si le problème est compliqué à résoudre. Les méthodes approchées, encore appelées heuristiques, permettent quant à elles d'obtenir rapidement une solution approchée, mais qui n'est donc pas toujours optimale.
- Le problème du voyageur de commerce est toujours d'actualité dans la recherche en informatique, étant donné le nombre important de problèmes réels auxquels il correspond.

5 | Projet calcul scientifique :

- Les importantes étapes historique de la résolution de problème de voyageur de commerce en accord avec le nombre des villes traitées entre 1954 et 2001 ; présentées ci-dessous :

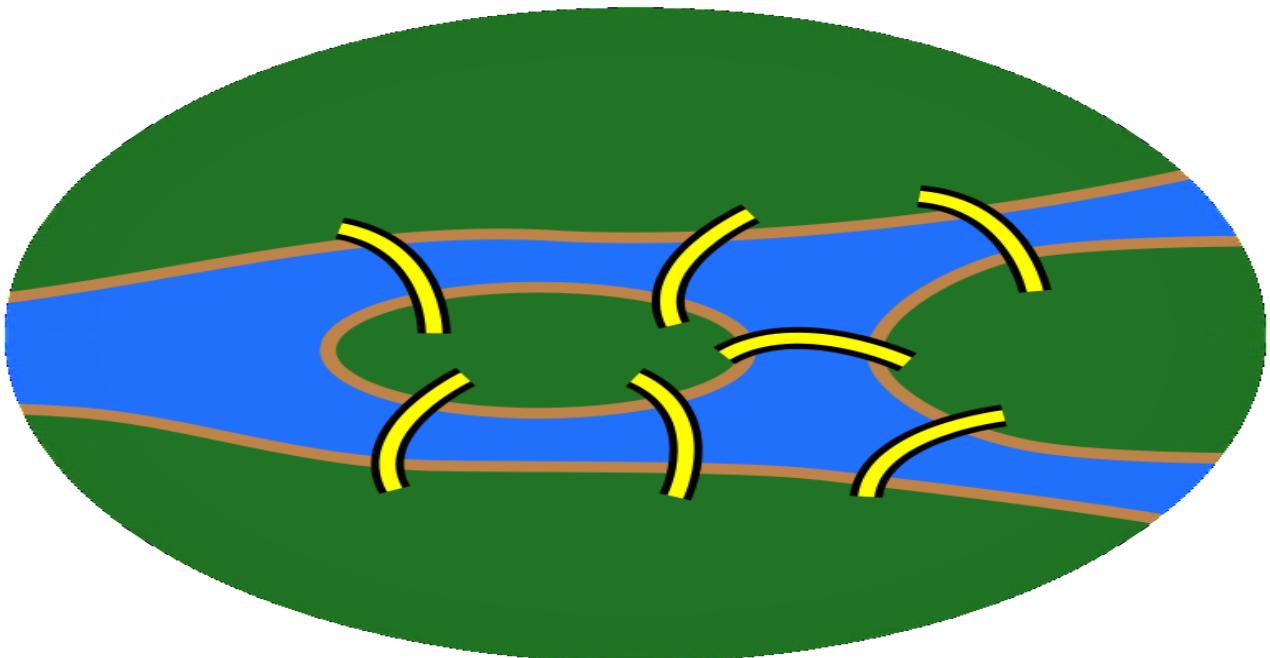
Année	Auteurs	Nombre de villes
1954	G.Dantzig, R.Fulkerson et S.Jhonson	49
1971	M. Held et R.M. Karp	64
1975	P.M.Camerini, L.Fratta et F.Maffioli	100
1977	M.Grotschel	120
1980	H.Crowder et M.W.Padberg	318
1987	M.Padberg et G.Rinaldi	532
1987	M. Grotschel et O.Holland	666
1987	M.Padberg et G.Rinaldi	2392
1994	D.Applegate, R.Bixby, V.Chvatal et W.Cook	7397
1998	D.Applegate, R.Bixby, V.Chvatal et W.Cook	13509
2001	D.Applegate, R.Bixby, V.Chvatal et W.Cook	15112

Chapitre I

Notion de base sur la théorie des graphes :

Puisqu'on va parler beaucoup sur les graphes dans le traitement de notre projet, on a décidé de commencer par la définition des notions fondamentales des graphes.

a. Historique :

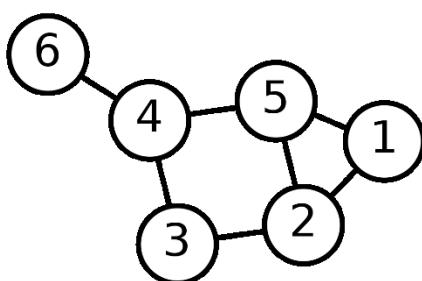


Un article du mathématicien suisse Leonhard Euler, présenté à l'Académie de Saint-Pétersbourg en 1735 puis publié en 1741, traitait du problème des sept ponts de Königsberg, ainsi que schématisé ci-dessous. Le problème consistait à trouver une promenade à partir d'un point donné qui fasse revenir à ce point en passant une fois et une seule par chacun des sept ponts de la ville de Königsberg. Un chemin passant par toute arête exactement une fois fut nommé chemin eulérien, ou circuit eulérien s'il finit

là où il a commencé. Par extension, un graphe admettant un circuit eulérien est dit *graphe eulérien*, ce qui constitue donc le premier cas de propriété d'un graphe. Euler avait formulé² qu'un graphe n'est eulérien que si chaque sommet a un nombre pair d'arêtes. L'usage est de s'y référer comme *théorème d'Euler*, bien que la preuve n'en ait été apportée que 130 ans plus tard par le mathématicien allemand Carl Hierholzer. Un problème similaire consiste à passer par chaque sommet exactement une fois, et fut d'abord résolu avec le cas particulier d'un cavalier devant visiter chaque case d'un échiquier par le théoricien d'échecs arabe Al-Adli dans son ouvrage *Kitab ash-shatranj* paru vers 840 et perdu depuis. Ce problème du cavalier fut étudié plus en détail au XVIII^e siècle par les mathématiciens français Alexandre-Théophile Vandermonde, Pierre Rémond de Montmort et Abraham de Moivre; le mathématicien britannique Thomas Kirkman étudia le problème plus général du parcours où on ne peut passer par un sommet qu'une fois, mais un tel parcours prit finalement le nom de chemin hamiltonien d'après le mathématicien irlandais William Rowan Hamilton, et bien que ce dernier n'en ait étudié qu'un cas particulier. On accorde donc à Euler l'origine de la théorie des graphes parce qu'il fut le premier à proposer un traitement mathématique de la question, suivi par Vandermonde.

b.Présentation :

La théorie des graphes est une discipline des mathématiques qui étudie les *graphes*. Ces derniers sont des modèles abstraits de dessins de réseaux reliant des objets. Ces modèles sont constitués par la donnée de « points », appelés *nœuds* ou *sommets*, et de « liens » entre ces points qui s'appellent souvent arêtes.



Les algorithmes élaborés pour résoudre des problèmes concernant les objets de cette théorie ont de nombreuses applications dans tous les domaines liés à la notion de réseau (réseau social, réseau informatique, télécommunications, etc.) et dans bien d'autres domaines (par exemple génétique). De grands théorèmes difficiles, comme le théorème des quatre couleurs, le théorème des graphes parfaits, ou encore le théorème de Robertson-Seymour, ont contribué à asseoir cette matière auprès des mathématiciens.

c. Définition des graphes et vocabulaire :

Pour définir un graphe général, il faut une fonction γ (gamma) qui associe à chaque arête une paire de sommets. Ainsi, un graphe est un triplet $G = (S, A, \gamma)$ avec S est l'ensemble des nœuds ou sommet (vertices en anglais), A est l'ensemble des arêtes (edges en anglais) et $\gamma : A \rightarrow S \times S$ la fonction qui donne les deux nœud liés par un arête. Toutefois l'usage veut que l'on note simplement $G = (S, A)$.

- Un graphe non orienté G est un couple (S, A) où S est un ensemble fini (l'ensemble des sommets) et A l'ensemble des paires constituées de sommets (les arêtes).
- Une chaîne d'un graphe non orienté G est une séquence de sommets de G
- Un graphe non orienté est connexe si chaque paire de sommet est reliée par une chaîne.
- Un graphe non orienté acyclique est appelée forêt, un graphe non orienté connexe acyclique est appelé un arbre.
- Un cycle hamiltonien d'un graphe non orienté G est un cycle où chaque sommet de G n'est visité qu'une et une seule fois.
 - Une chaîne $< v_i >_{1 \leq i \leq n}$ est un cycle si et seulement si ($i \geq 3$, $v_0 = v_n$ et $\forall i \neq j$, $v_i \neq v_j$).

d. Graphe Eulérien :

En théorie des graphes, on dit d'un graphe non orienté qu'il est « eulérien » en référence à Euler (la plupart des mathématiciens écrivent « Eulérien » à cause de l'usage anglo-saxon) s'il a la propriété suivante :

On peut distinguer une extrémité initiale et une extrémité finale de chaque arête, et ordonner l'ensemble des arêtes du graphe de telle façon que l'extrémité finale d'une

arête corresponde à l'extrémité initiale de l'arête qui lui succède dans l'ordre (où la première arête de l'ordre succède à la dernière).

Cette propriété est équivalente au fait que l'on peut « parcourir » le graphe en partant d'un sommet quelconque et en empruntant exactement une fois chaque arête pour revenir au sommet de départ, il admet donc un **cycle eulérien**. Un tel graphe a alors la propriété qu'il correspond à un dessin qu'on peut tracer sans lever le crayon.

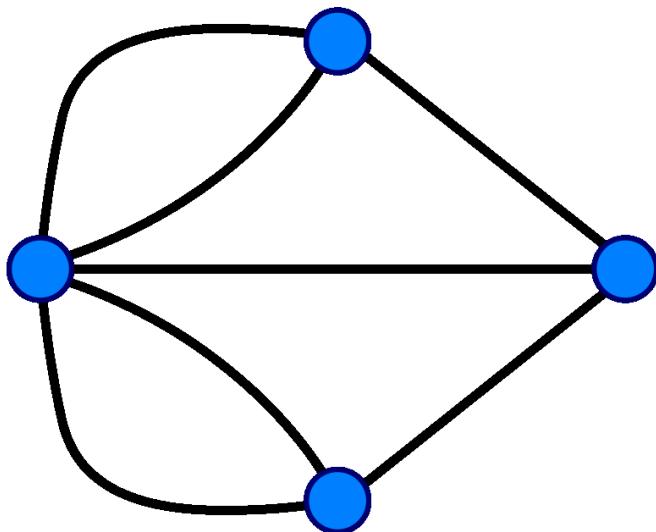
i. Théorème d'Euler :

La preuve du théorème d'Euler ci-dessous fut en fait publiée par Carl Hierholzer en 1873, on l'appelle donc aussi le théorème d'Euler-Hierholzer :

Enoncé du théorème (1736) — Un graphe connexe est eulérien si et seulement si chacun de ses sommets est incident à un nombre pair d'arêtes.

ii. Preuve :

La nécessité est pratiquement immédiate (l'argument est le même que pour la résolution du problème des sept ponts de Königsberg).



La suffisance n'est pas non plus beaucoup plus dure.

Rappelons d'abord quelques définitions :

- Le *degré* d'un sommet est le nombre d'arêtes incidentes au sommet ;

- Un *parcours* est une suite d'arêtes telle que (i) pour chaque arête de la suite on peut distinguer une extrémité initiale et une terminale, (ii) l'extrémité terminale d'une arête est aussi l'extrémité initiale de l'arête qui lui succède dans la suite (et la première arête succède à la dernière) ;
- Un *circuit* est un parcours non vide tel qu'aucun sommet n'est l'extrémité initiale (terminale) de plus d'une arête.

La condition suffisante du théorème d'Euler-Hierholzer s'appuie principalement sur les trois faits suivants :

Lemme 1 :

Si tous les degrés sont pairs et non tous nuls, alors il existe un circuit ;

Lemme 2 :

Un parcours est une union de circuits disjoints – au niveau des arêtes, et non des sommets ;

Lemme 3 :

Si l'on retire les arêtes d'un parcours, alors les degrés pairs restent pairs.

Supposons maintenant que chaque sommet a un degré pair et qu'il n'existe pas de parcours contenant toutes les arêtes. Si l'on considère un parcours avec un nombre maximum d'arêtes et que l'on retire ensuite les arêtes du parcours du graphe, par (3), les degrés restent pairs. D'où, par (1), l'existence d'un circuit disjoint de notre parcours maximum. Mais, par (2), l'union de notre parcours et du circuit forme un autre parcours avec plus d'arêtes, ce qui contredit l'hypothèse de maximalité du parcours initial. Cette contradiction implique donc le théorème.

Chapitre II

Etude mathématique du problème :

Ce problème a la particularité d'être impossible à résoudre de manière exacte en un temps raisonnable. Voici un tableau qui illustre bien le propos, en considérant qu'un trajet s'évalue en 1 microseconde (ce qui est déjà très rapide)

Nombre de villes	Nombre de possibilités	Temps de calcul
5	12	12 microsecondes
10	181440	0,18 secondes
15	43 milliards	12 heures
20	$60 \cdot 10^{15}$	1928 ans
25	$310 \cdot 10^{21}$	9,8 milliards d'années

Si N est le nombre de villes, le nombre de possibilités est $(N - 1)! / 2$; la croissance est donc exponentielle. Il est alors nécessaire de s'intéresser à des méthodes d'approximations qui nous donnent un chemin relativement court en un temps raisonnable, le calcul exhaustif étant à ce stade impossible (il prendrait sans doute des milliards de milliards d'années).

Formulation du problème du voyageur de commerce :

Pour commencer, nous avons considéré la formulation du problème du voyageur de commerce. Cette formulation a été proposée pour la première fois par Dantzig, Fulkerson et Jonhson en 1954, nous pouvons la décrire de la façon suivante : Soit x_{ij} une variable binaire telle que $x_{ij}=1$ si l'arc (i,j) appartient à la solution du PVCA et $x_{ij}=0$ sinon. Donc on veut :

$$\text{Minimiser} \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

Sous les contraintes :

$$\sum_{v_j \in V} x_{ij} = 1 \quad \forall v_i \in V \text{ et } v_i \neq v_j \quad (2)$$

$$\sum_{v_j \in V} x_{ji} = 1 \quad \forall v_i \in V \text{ et } v_i \neq v_j \quad (3)$$

$$\sum_{v_i, v_j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V, 3 \leq |S| \leq (n-3) \quad (4)$$

$$x_{ij} \in \{0,1\} \quad \forall v_i, v_j \in V \quad (5)$$

Les contraintes (2) et (3) assurent qu'un sommet n'est visité qu'une fois par cycle : on y arrive une et une seule fois (contrainte (2)) et on en part une et une seule fois (contrainte (3)). La contrainte (5), appelée contrainte d'intégrité, détermine le domaine des variables x_{ij} . La contrainte (4) est une contrainte d'élimination des sous tours avec S est un sous-ensemble de sommet de V . Par exemple, la forme du cycle présenté par (figure 2.1) n'est pas permise par cette contrainte. Ce cycle n'est pas hamiltonien, donc il ne peut pas être une solution du PVC.

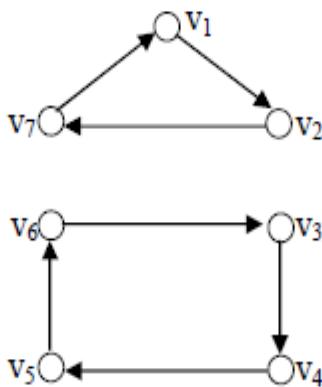


Figure : Exemple de sous tours

Grotschel et Patberg , ont prouvé que cette contrainte est une facette du polytope de l'enveloppe convexe définie par les contraintes de (2) à (5). Si la cardinalité de S est égale à 2, la somme $\sum_{v_i, v_j \in S} x_{ij}$ vaut au maximum 1

donc La contrainte est inutile. Cette contrainte génère un nombre exponentiel de contraintes selon le nombre des sommets à résoudre.

En 1960, Miller, Tucker et Zemlin (M.T.Z) ont proposé une contrainte d'élimination des sous tours qui génère un nombre polynomial de contraintes, elle peut être exprimée comme il suit [5] :

$$u_i - u_j + (n-1)x_{ij} \leq (n-2) \quad \forall v_i, v_j \in V - \{v_1\} \quad (6)$$

$$1 \leq u_i \leq n-1 \quad \forall v_i \in V - \{v_1\} \quad (7)$$

où u_i désigne l'ordre du sommet i dans le cycle. Cette contrainte assure qu'il y a uniquement une solution qui correspond au cycle réalisable. Si nous appliquons cette contrainte au sous-cycle composé par $\{v_3, v_4, v_5, v_6\}$ de l'exemple précédent (figure), nous obtenons les contraintes qui mènent à une contradiction :

$$\begin{aligned} (u_3 - u_4) + (n-1)x_{34} &\leq (n-2) \\ (u_4 - u_5) + (n-1)x_{45} &\leq (n-2) \\ (u_5 - u_6) + (n-1)x_{56} &\leq (n-2) \\ (u_6 - u_3) + (n-1)x_{63} &\leq (n-2) \end{aligned}$$

En sommant ces contraintes nous obtenons $(n-1) < (n-2) !!!$

Ce sous tour n'est pas permis par cette contrainte, mais elle représente une faiblesse. Elle n'est en effet pas une facette pour le polytope de l'enveloppe convexe des solutions réalisables définie par les contraintes (2), (3), (6), (7) et (5). Pour cela, cette contrainte a été amélioré par Laporte et Desrochers en 1990 [1]. Leur proposition est la contrainte suivante et qui représente une inégalité valide pour le PVC :

$$u_i - u_j + (n-1)x_{ij} + (n-3)x_{ji} \leq (n-2) \quad \forall v_i, v_j \in V - \{v_1\} \quad (8)$$

Preuve :

Cette contrainte est obtenue en utilisant la technique de *lifting* [12,13] sur la contrainte (6). On considère la contrainte d'élimination des sous tours de M.T.Z [5] et on lifte la variable x_{ji} .

$$u_i - u_j + (n-1)x_{ij} + \alpha_{ji}x_{ji} \leq (n-2) \text{ avec } \alpha_{ji} = 0 \quad (*)$$

Le processus de *lifting* prend la plus grande valeur possible de α_{ji} pour que (*) demeure une inégalité valide.

Deux cas qui peuvent se présenter : $x_{ji} = 0$ et $x_{ji} = 1$.

Cas1 : $x_{ji} = 0$, (*) est satisfaite pour n'importe quel α_{ji} .

Cas2 : $x_{ji} = 1 \Rightarrow x_{ij} = 0$ (pour $n > 2$)

$$\Rightarrow u_i - u_j = 1$$

donc :

$$1 + \alpha_{ji} \leq n - 2$$

$$\alpha_{ji} \leq n - 3$$

Puisque α_{ji} prend la plus grande valeur on prend $\alpha_{ji} = n - 3$, d'où la contrainte valide (6) du PVC.

Pour la suite nous allons prendre en considération la formulation du PVCA composée par les contraintes (1), (2), (3), (8), (7) et (5).

$$\text{Minimiser} \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

Sous les contraintes :

$$\sum_{v_j \in V} x_{ij} = 1 \quad \forall v_i \in V \text{ et } v_i \neq v_j \quad (2)$$

$$\sum_{v_j \in V} x_{ji} = 1 \quad \forall v_i \in V \text{ et } v_i \neq v_j \quad (3)$$

$$u_i - u_j + (n-1)x_{ij} + (n-3)x_{ji} \leq (n-2) \quad \forall v_i, v_j \in V - \{v_1\} \quad (8)$$

$$1 \leq u_i \leq n-1 \quad \forall v_i \in V - \{v_1\} \quad (7)$$

$$x_{ij} \in \{0,1\} \quad \forall v_i, v_j \in V \quad (5)$$

Chapitre III

Implémentation numérique du problème :

Nous avons vu que le problème de voyageur de commerce admet une solution, donc pour visiter plusieurs villes il existe le plus court chemin à suivre, et cela est sûr et démontré grâce à l'évolution de l'informatique. Mais vu qu'on a seulement traité la possibilité d'existence d'un tel chemin, alors la démarche à suivre pour l'effectuer sur une carte donnée en particulier celle du MAROC n'est pas si claire et demeure ambiguë. Dans cette partie alors, on va s'intéresser à étudier les différentes méthodes ou algorithmes à suivre permettant de traverser les villes en question tout en respectant la contrainte du plus court chemin. Afin de mettre en lumière les résultats de ces algorithmes, on a effectué leur implémentation numérique dans un programme équipé d'une interface permettant d'interagir avec les résultats en temps réels, deux algorithmes principaux vont être étudié, chacun va améliorer le résultat pour arriver au meilleur :

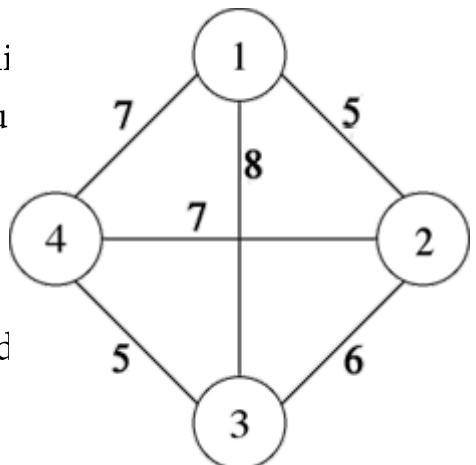
- 1. Algorithme Dijkstra**
- 2. Algorithme d'amélioration 2-opt**

On a appliqué ces deux algorithmes sur la carte du MAROC, mais on peut les appliquer sur n'importe quel pays du monde.

Le programme permettant la simulation des deux algorithmes sur les villes marocaines a été en langage C, en utilisant la bibliothèque SDL2 pour réaliser la partie graphique, l'interface est intuitive et simple à comprendre, c'est-à-dire qu'il est facile de naviguer sur le programme voir les différents résultats.

III.1 Modélisation numérique du problème :

Le problème du voyageur de commerce peut être modélisé à l'aide d'un graphe constitué d'un ensemble de sommets et d'un ensemble d'arêtes. Chaque sommet représente une ville, une arête symbolise le passage d'une ville à une autre, et on lui associe un poids pouvant représenter une distance, un temps de parcours ou encore un coût. Ci-contre, un exemple de graphe à 4 sommets.



Résoudre le problème du voyageur de commerce revient à trouver dans ce graphe un cycle passant par tous les sommets une unique fois (un tel cycle est dit « hamiltonien ») et qui soit de longueur minimale. Pour le graphe ci-contre, une solution à ce problème serait le cycle 1, 2, 3, 4 et 1, correspondant à une distance totale de 23. Cette solution est optimale, il n'en existe pas de meilleure.

Comme il existe une arête entre chaque paire de sommets, on dit que ce graphe est « complet ». Pour tout graphe, une matrice de poids peut être établie. En lignes figurent les sommets d'origine des arêtes et en colonnes les sommets de destination ; le poids sur chaque arête apparaît à l'intersection de la ligne et de la colonne correspondantes. Pour notre exemple, cette matrice est la suivante :

0	5	8	7
5	0	6	7
8	6	0	5
7	7	5	0

Dans cet exemple, le graphe est « non orienté », c'est-à-dire qu'une arête peut être parcourue indifféremment dans les deux sens, cela explique que la matrice soit symétrique. Cette symétrie n'est pas forcément respectée dans le cas d'un graphe orienté. Il existe alors deux catégories de problèmes : le cas symétrique (le poids de l'arc

17 | Projet calcule scientifique :

du sommet X vers Y est égal au poids de l'arc du sommet Y vers X) et le cas asymétrique (le poids de l'arc du sommet X vers Y peut être différent du poids de l'arc du sommet Y vers X). Mais dans notre cas c'est le cas symétrique puisque la distance entre deux villes est fixe dans les deux sens.

Cas de la carte du Maroc :



Est c'est voila notre base de donnée sur les distances entre les villes marocaines :

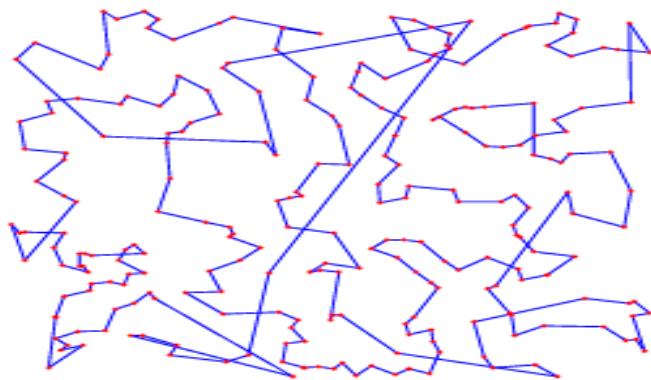
DISTANCES INTERVILLES

	AGADIR	AL HOCEIMA	BENI MELLAL	CASABLANCA	DAKHLA	EL JADIDA	ERRACHIDIA	ESSAOUIRA	FES	FIGUIG	KENITRA	KHENIFRA	KHOURIBGA	LAAYOUNE	LAGOUIRA	MARRAKECH	MEKNES	NADOR	OUARZAZATE	OUJDA	RABAT	SAFI	SETTAT	TANGER	TAN-TAN	TAZA	TETOUAN
AGADIR																											
AL HOCEIMA	1091																										
BENI MELLAL	467	564																									
CASABLANCA	511	536	210																								
DAKHLA	1173	2264	1640	1684																							
EL JADIDA	417	632	271	99	1590																						
ERRACHIDIA	681	616	375	545	1654	506																					
ESSAOUIRA	173	887	370	351	1346	252	745																				
FES	756	275	299	209	1920	388	364	640																			
FIGUIG	1076	669	770	920	2249	1001	395	1081	719																		
KENITRA	598	435	122	131	1769	360	292	499	160	687																	
KHENIFRA	642	400	300	299	1615	230	400	492	166	675	254																
KHOURIBGA	507	614	90	120	160	181	425	410	333	820	169	245															
LAAYOUNE	649	1740	1116	1160	524	1066	1330	822	1396	1725	1245	1291	1156														
LAGOUIRA	1645	2736	2112	2156	472	2062	2336	1819	2392	2721	2241	2287	2152	996													
MARRAKECH	273	758	194	238	1446	197	510	176	483	905	234	369	234	922	1918												
MEKNES	740	335	278	229	1913	328	346	580	60	741	153	130	322	1309	2389	467											
NADOR	1035	175	628	628	2360	727	510	9579	339	515	323	497	672	1736	2732	822	399										
OUARZAZATE	375	922	398	442	1548	399	306	380	667	701	527	573	488	1024	2020	204	652	816									
OUJDA	1099	298	632	632	2272	731	514	963	343	326	503	507	676	1748	2744	626	403	104	620								
RABAT	602	416	250	91	1775	190	482	442	198	877	40	269	205	1251	2247	321	138	535	528	541							
SAFI	294	792	351	256	1467	157	683	129	545	1078	480	387	328	943	1939	157	406	894	361	888	347						
SETTAT	439	608	157	72	1612	117	512	296	361	907	238	203	69	1008	2084	166	292	969	370	698	157	201					
TANGER	880	323	536	369	2053	468	608	720	303	988	620	238	483	1529	2525	598	267	1086	811	609	278	625	433				
TAN-TAN	331	1422	798	842	749	1012	504	1067	1407	927	976	838	318	1314	504	1071	1426	705	1430	938	625	770	1211				
TAZA	676	173	409	409	2049	508	760	120	599	206	453	280	1525	2521	603	100	219	790	223	318	665	474	423	1217	1420		
TETOUAN	892	278	536	365	2065	484	604	706	261	981	411	254	499	1541	2537	675	288	437	280	555	294	641	460	57	1223	370	

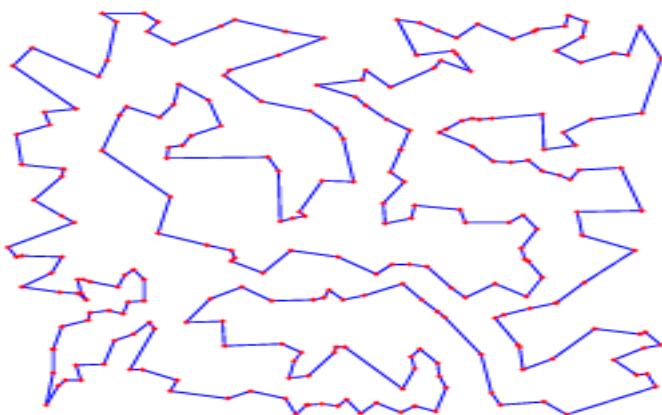
III.2 Dijkstra et 2-opt pour trouver la solution :

Avant de découvrir le principe de Dijkstra et celui de 2-Opt on va d'abord montrer l'influence de la combinaison de ces méthodes pour avoir la solution ; Ces deux captures qui suivent montre l'importance de cette combinaison.

Méthode naïve des plus proches voisins



Plus proches voisins + optimisation locale 2-Opt



III.3 Dijkstra :

1 Principe :

L'algorithme dû à **Dijkstra** est basé sur le principe suivant :

Si le plus court chemin reliant E à S passe par les sommets s_1, s_2, \dots, s_k alors, les différentes étapes sont aussi les plus courts chemins reliant E aux différents sommets s_1, s_2, \dots, s_k .

On construit de proche en proche le chemin cherché en choisissant à chaque itération de l'algorithme, un sommet s_i du graphe parmi ceux qui n'ont pas encore été traités, tel que la longueur connue provisoirement du plus court chemin allant de E à s_i soit la plus courte possible.

Initialisation de l'algorithme :

Étape 1 :

On affecte le poids 0 au sommet origine (E) et on attribue provisoirement un poids ∞ aux autres sommets.

Répéter les opérations suivantes tant que le sommet de sortie (s) n'est pas affecté d'un poids définitif

Étape 2 :

Parmi les sommets dont le poids n'est pas définitivement fixé choisir le sommet X de poids p minimal.

Marquer définitivement ce sommet X affecté du poids $p(X)$.

Étape 3 :

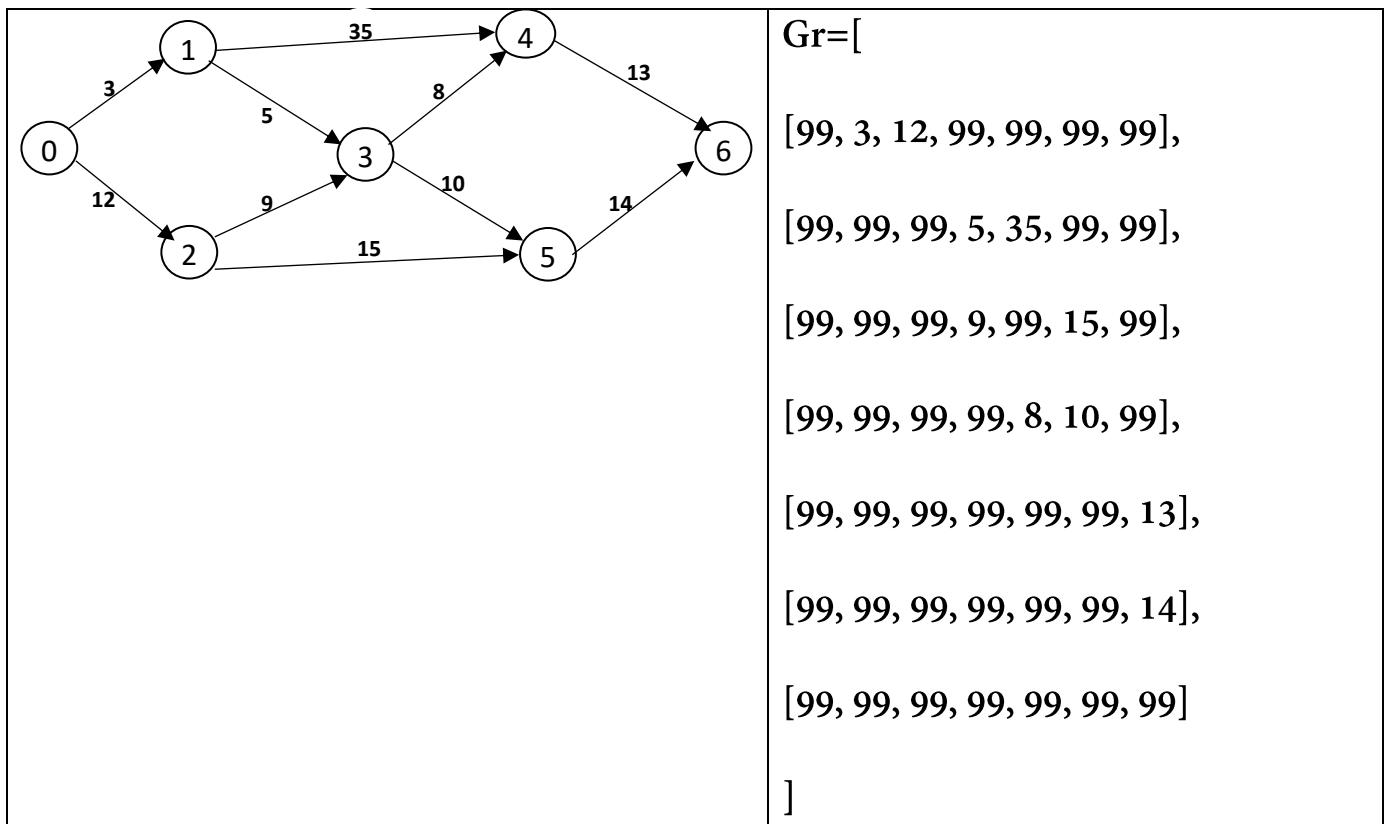
Pour tous les sommets Y qui ne sont pas définitivement marqués, adjacents au dernier sommet fixé X :

- Calculer la somme s du poids de X et du poids de l'arête reliant X à Y .
- Si la somme s est inférieure au poids provisoirement affecté au sommet Y , affecter provisoirement à Y le nouveau poids s et indiquer entre parenthèses le sommet X pour se souvenir de sa provenance.

Quand le sommet s est définitivement marqué

Le plus court chemin de E à S s'obtient en écrivant de gauche à droite le parcours en partant de la fin S .

2 Application :



>>>`dijkstra(Gr,0)`

([0, 0, 0, 1, 3, 3, 4] , [0, 3, 12, 8, 16, 18, 29])

III.4 Amélioration 2-opt :

1 Principe :

Le principe de l'algorithme « amélioration 2-opt » est assez simple et repose sur l'idée suivante :

- On génère un premier trajet
- On l'améliore jusqu'à ce qu'un critère d'arrêt ait été vérifié.

Génération du premier trajet : On peut générer le trajet de telle sorte qu'à la i-ème étape, le choix de la $(i + 1)$ -ème ville est la plus proche voisine (Dijkstra) de la i-ième ville parmi les villes à parcourir.

Amélioration du trajet : Une amélioration possible est de supprimer tout croisement. En s'appuyant sur l'inégalité du parallélogramme, prenons le cas d'un trajet ou un chemin direct relie A à B, un chemin relie B à C et ensuite un chemin relie C à D (exemple illustré dans la figure 5). On voit que si $((AB) + (CD) > (AC) + (BD))$ alors il y a présence de croisement et il apparaît plus efficace alors de passer de A à C, de relier C à B puis de passer de B à D.

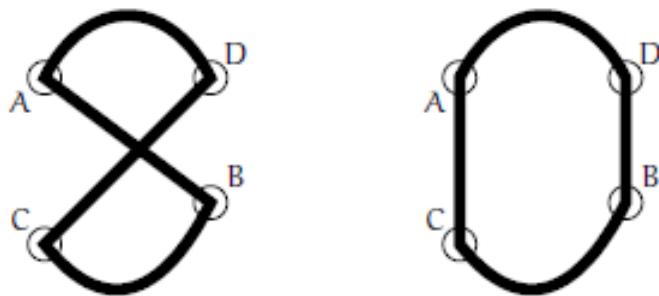


FIGURE 5 – un exemple de croisement

Critère d'arrêt : Un critère d'arrêt possible est que la fonction d'amélioration définie précédemment ne modifie plus le trajet. Autrement dit dès qu'il n'y a plus de croisement dans le trajet.

2 Application :



Temps : 0.02 sec

Parcours : 1082

3 Algorithme :

```

Données : L % contient les villes à parcourir avec des informations sur les distances entre les villes
Résultat : L' : liste % contient la liste du parcours de villes à effectuer
début
    L' ← Générer-trajet-initial S
    B1 ← true
    tant que B1 faire
        L'' ← amélioration L'
        si Trajet L'' = Trajet L' alors
            | B1 ← false
        fin
        si Trajet L'' < Trajet L' alors
            | L' ← L''
        fin
    fin
fin

```

Algorithme 2: Algorithme 2-Opt

Complexité en $O(l^2)$

Chapitre III

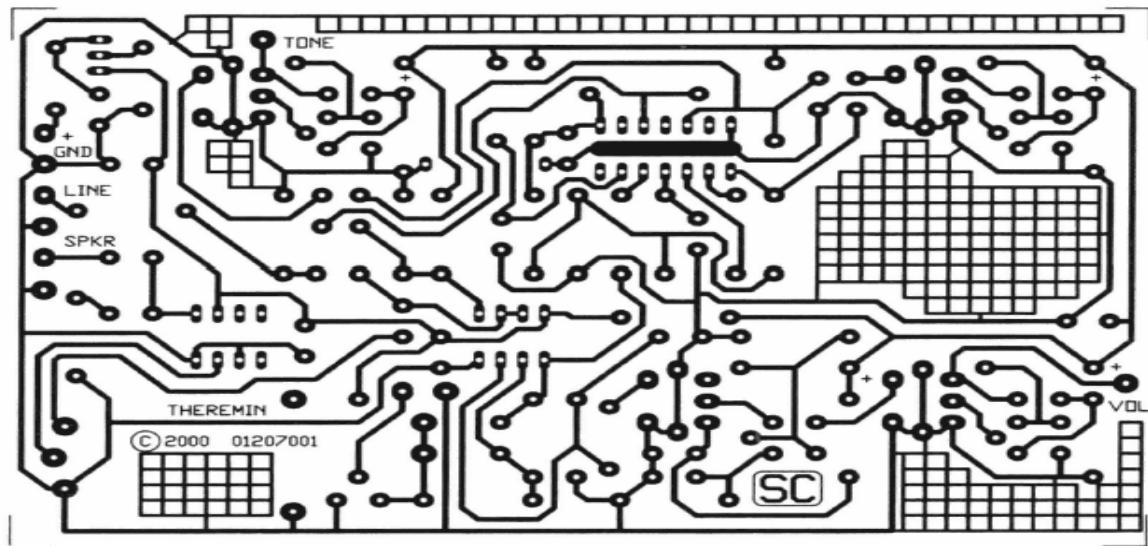
Champs d'application :

Le problème du voyageur du commerce a de nombreuses applications, et a d'ailleurs été motivé par des problèmes concrets, par exemple le parcours des bus scolaires. Un premier type d'application classique est bien sûr dans la logistique, par exemple pour la poste, la distribution de repas à domicile, l'inspection d'installation, etc. On peut aussi optimiser les mouvements de machines, par exemple, pour minimiser le temps total que met une fraiseuse à commande numérique pour percer n points dans une plaque de tôle, ou minimiser les mouvements des grands télescopes (qui sont très lents). On peut citer des applications plus surprenantes, par exemple : en biologie, le problème aide au séquençage du génome (pour relier des petits fragments en des chaînes plus grandes), et en production il est utilisé pour le test des circuits imprimés.

1. Industrie électronique

L'industrie de l'électronique utilise des chaînes de fabrications automatisées pour réaliser les cartes électroniques comme les cartes mères d'ordinateurs. Après le gravage du circuit proprement dit il y a une phase de perçage des trous permettant d'enficher les composants, puis de les souder en place. Nous allons nous intéresser à la phase de perçage. Il y a souvent plusieurs centaines de trous à percer par une perceuse automatique se déplaçant de trou en trou. Les industriels ont cherché à minimiser le temps de déplacement à vide de la tête de perçage (time is money!). Ce temps est lié à la distance séparant les trous. Ce problème est similaire au problème mathématique dit du

"voyageur de commerce" (tsp : travelling salesman problem).



2. Sécurité et maintenance

De nos jours la sécurité est devenue une obligation et utilise les dernières technologies et les compétences de la gestion. Les vigiles appelés « agent de sécurité ou de surveillance », sont de plus en plus employés, ceci afin de faire face et de palier à un manque de sécurité flagrant aux vues des actes de vandalisme et de malveillance commis auprès des commerçants, entreprises et même des particuliers . Les compagnies de sécurité offrent une large variété de services pour satisfaire aux besoins de leurs clients, tels que la surveillance des banques, des centres commerciales.... La ronde du veilleur de nuit est une application du PVCM . Un veilleur doit faire sa tournée en optimisant la longueur de son parcours et par suite la durée de sa tournée et en tenant compte du degré d'importance attribué à chaque local qu'il doit visiter nh fois par nuit. L'ensemble des sommets de même couleur correspond à un même local. La cardinalité d'un tel ensemble désigne le nombre des visites à effectuer pour cet endroit.

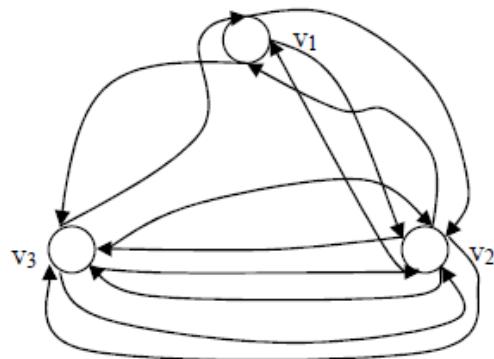


Figure 1.2 : Présentation d'un PVCM sous forme initiale

La détermination anticipée de la ronde du veilleur de nuit, permet au veilleur d'établir sa tournée normalement et d'éliminer les failles de la sécurité qui peuvent survenir pendant la nuit. En effet, la fixation des limites h_a et h_b dépend directement de l'importance et du taux de sécurité attribué au sommet de type h .

Conclusion et perspectives :

Le problème de voyageur de commerce s'avère être parmi les problèmes les plus célèbres de mathématiques combinatoires. Plusieurs mathématiciens l'ont abordé, et plusieurs d'entre eux ont abouti à des résultats. Jusqu'à l'instant, les démonstrations approuvées du problème sont assistées par ordinateur. De plus, l'importance de voyageur de commerce dépasse la théorie au concret, puisqu'il sert à résoudre des problèmes d'affectation et de planning. Ainsi, en traitant ce sujet, nous avons voulu mettre en pratique ce que nous avons eu d'apprentissage lors de cette première année, et aussi de faire un travail collectif où chacun des membres de l'équipe a contribué à la réalisation de cette étude théorique-numérique du problème.

Bibliographie :

- <http://lycees-reaumur-buron.fr/immatriculation-voiture-maroc-par-ville/>
- https://fr.wikipedia.org/wiki/Probl%C3%A8me_du_voyageur_de_commerce
- <http://www.btb.termiumplus.gc.ca/tpv2alpha/alpha-fra.html?lang=fra&srctxt=travelling+salesman+problem>
- <https://interstices.info/le-probleme-du-voyageur-de-commerce/>
- <http://villemin.gerard.free.fr/LogForm/GrVoyage.htm>
- https://www.gerad.ca/Sebastien.Le.Digabel/MTH6311/8_applications_2.pdf