British Journal of Environmental Sciences
Vol.11, No.2, pp.,46-66, 2023
Print ISSN: 2055-0219(Print)
Online ISSN: 2055-0227(online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development–UK

Implémentation des Parcours eulériens basés sur le modèle du postier chinois : Application aux problèmes de ramassage des ordures dans les rues de la Ville de Kananga

[Implementation of Eulerian Paths based on the model of the Chinese postman: Application to the problems of garbage collection in the streets of the City of Kananga].

[1]Dr. Sr. Marie-Alice TSHIELA Nkuna

[1]Professeure attachée à la faculté d'Informatique de l'Université Notre Dame du Kasayi (U.KA.), Ville de Kananga, Kasaï Central (RDC)

doi: https://doi.org/10.37745/bjes.2013/vol11n24666 Published June 5, 2023

Citation : Nkuna M.T. (2023) Implementation of Eulerian Paths based on the model of the Chinese postman: Application to the problems of garbage collection in the streets of the City of Kananga, *British Journal of Environmental Sciences*,Vol.11, No.2, pp.,46-66

RESUME : *La théorie de graphe joue aujourd'hui un rôle très important dans la modélisation et la résolution de beaucoup de problèmes. La complexité de ce sujet est de taille, d'autant plus que nous devons parler en même temps du postier chinois et du parcours eulérien. S'il reste vrai que la préoccupation est la recherche du chemin qui minimise le coût, nous comprenons qu'il nous faudra impérativement glaner sur l'algorithme de Fleury, sans manquer de déboucher sur les possibles conditions de la réalisation des rêves d'Euler sur la ville de Kananga, dont le réseau routier est plus que complexe dans l'enchevêtrement de ses artères. Dans cet article nous cherchons à voir comment on peut résoudre le problème du ramassage des ordures en passant par chaque carrefour une et une seule fois minimisant ainsi le coût et le temps.*

MOTS-CLES *:* implémentation, parcours eulérien, modèle, postier chinois, application, problème, ramassage, ordures, rue, ville, Kananga.

ABSTRACT: *Graph theory today plays a very important role in modeling and solving many problems. The complexity of this subject is considerable, especially since we must speak at the same time of the Chinese postman and the Eulerian journey. If it remains true that the concern is the search for the path which minimizes the cost, we understand that it will be imperative for us to glean from Fleury's algorithm, without failing to lead to the possible conditions for the realization of Euler's dreams. On the city of Kananga, whose road network is more than complex in the tangle of its arteries? In this article we seek to see how we can solve the problem of garbage collection by going through each intersection once and only once, thus minimizing cost and time.*

KEYWORDS*:* implementation, eulerian path, model, Chinese postman, application, problem, pickup, garbage, street, city, Kananga.

INTRODUCTION

La présente étude porte sur l'application de la théorie de graphe (orienté) **(3)** à la possibilité d'un parcours sans faute dans le travail de ramassage des ordures dans les rues de la ville de Kananga en se basant sur le problème du postier chinois. *C'est plutôt le problème eulérien qui est attaché à la théorie des graphes*, c'est-à-dire à la recherche d'une chaîne ou d'un cycle passant exactement une fois par chaque arête ou par chaque sommet.

Il s'agit pour nous de savoir dans quelle mesure on peut parcourir les rues de la ville de Kananga, en passant par chaque carrefour une et une seule fois, en vue de résoudre le problème d'insalubrité par le ramassage des ordures de ménage, **(6)** et cela en adoptant une méthode permettant de trouver la trajectoire la plus courte à la manière dont procède le postier chinois. En cherchant à trouver une chaîne ou un cycle passant exactement une fois par chaque arête (chaque rue), l'exemple du postier chinois nous renvoie à la méthode du parcours eulérien et nous propose une façon de parcourir les rues de la ville de Kananga, en passant une seule fois dans chaque rue **(10)**. L'avantage sera celui de minimiser la longueur ou la distance du parcours. Dans le cadre d'un graphe qui n'est pas forcément eulérien, on trouve la solution à ce genre de problème dans les organisations de tournées de distribution de courriers, de ramassage d'ordures, d'inspection de réseaux de distribution, etc.

Dans l'objectif de notre étude, trois points sont retenus pour structurer l'ensemble de l'analyse. Le premier point portera sur le problème du parcours eulérien, en définissant les concepts liés à cette théorie permettant de spécifier les *conditions nécessaires et suffisantes* (CNS) pour qu'un graphe soit eulérien **(12)**. On aboutira aussi à la régénérescence de l'algorithme en analysant sa complexité sur un exemple numérique. Le deuxième point s'attardera sur la résolution du problème du postier chinois en s'appuyant sur les théorèmes de la résolution des problèmes du parcours eulérien ainsi que celui du *problème T-joint de poids minimum.* Le troisième point se focalisera sur le ramassage des ordures ménagères dans la ville de Kananga ainsi que la mise en place de l'application pour faciliter cette tâche **(32)**.

Le parcours Eulérien associé à la théorie des graphes

Dans ce paragraphe, nous nous intéressons au problème du parcours eulérien, à travers les définitions des notions liées à la théorie des graphes, en vue de spécifier les critères d'un graphe qui soit Eulérien **(17)**. La démarche va aboutir au constat de la régénérescence des algorithmes sur des exemples numériques, à travers l'analyse de leur complexité. D'où il est nécessaire de commencer par définir les concepts-clés.

Définitions :

a. Graphes connexe et fortement connexe.

Soit G = (X, U) un graphe.

● G = (X, U) est connexe s'il existe au moins une chaîne entre une paire quelconque de sommets de G.

● G = (X, U) est fortement connexe s'il existe au moins une chaîne entre une paire quelconque de sommets de G.

British Journal of Environmental Sciences
Vol.11, No.2, pp.,46-66, 2023
Print ISSN: 2055-0219(Print)
Online ISSN: 2055-0227(online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development–UK

a. Chaînes et cycles eulériens

Soit G = (X, U) un graphe orienté.

- Un chemin eulérien est un chemin simple contenant tous les arcs de G.
- Un circuit eulérien est un chemin eulérien fermé.

Soit G = (X, U) un graphe non orienté.

- Une chaîne eulérienne est une chaîne simple qui contient toutes les arêtes de G.
- Un cycle eulérien est une chaîne eulérienne fermée.
- Un graphe eulérien est un graphe possédant un cycle ou circuit eulérien.

Exemple :

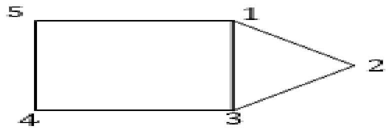


Fig 1 : *Graphe eulérien*

Problème du parcours eulérien

L'étude des problèmes eulériens remonte aux origines de la théorie des graphes. L'intérêt porté à ces problèmes s'explique par leurs nombreuses applications : tournées de distribution, tracé automatique sur ordinateur, problème d'ordonnancement d'atelier **(3)**, etc. Au 18ème siècle il y a eu un casse-tête populaire chez les habitants de Königsberg **(30)**. Une situation préoccupait la population de cette ville : la possibilité de se promener dans la ville en ne passant qu'une seule fois par chacun des sept ponts de Königsberg. C'est le célèbre mathématicien Euler, qui, le premier, tentera de proposer une solution à ce problème, en utilisant pour la première fois la notion de graphe **(12)**. En termes de graphe, le problème sera ainsi formulé : existe-t-il dans le multi-*graphe ci-dessus un cycle qui passe exactement une fois par toutes les arêtes ?*

Depuis lors, de son histoire, tout le monde s'accorde à considérer que la théorie des graphes est née en 1736 avec la communication d'Euler (1707-1783) dans laquelle il proposait une solution au célèbre problème des ponts de Königsberg **(9)**. Désormais, la théorie des graphes constitue aujourd'hui un corpus de connaissances très important **(6)**. Le langage des graphes est construit, à l'origine, sur des principes. Nombreuses méthodes, propriétés, et procédures ont été pensées ou trouvées à partir d'un schéma pour être ensuite formalisées et développées. Le problème consiste à déterminer s'il existe ou non une promenade dans les rues de Königsberg permettant, à partir d'un point de départ au choix, *de « passer une et une seule fois par chaque pont et de revenir à son point de départ, étant entendu qu'on ne peut traverser le Pregel qu'en passant sur les ponts »* **(30)**. Voici à quoi ressemblait la carte des ponts et comment on peut en arriver à un graphe.

Fig.1 : pont de la ville de Konigsberg à un graphe.

British Journal of Environmental Sciences
Vol.11, No.2, pp.,46-66, 2023
Print ISSN: 2055-0219(Print)
Online ISSN: 2055-0227(online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development–UK

Soit G un graphe non orienté. Une chaîne eulérienne est une chaîne qui passe une et une seule fois par toutes les arêtes de *G*. On définit les mêmes notions pour un *graphe orienté G* : un chemin ou un circuit eulérien est un chemin ou un circuit passant une et une seule fois par tous les arcs de *G*.

Enoncé du problème

1er cas : Graphe non orienté

Etant donné un graphe non orienté G = (X, U), existe-t-il une chaîne simple de G qui contienne toutes ses arêtes ?

2e cas : Graphe orienté

Etant donné un graphe orienté G = (X, U), existe-t-il un chemin simple de G qui contienne tous ses arcs ?

Comment savoir qu'un graphe non orienté (resp. orienté) G = (X, U) contient une chaîne eulérienne (resp. un chemin eulérien), ou un cycle (resp. un circuit) eulérien ?

Recherche de solution

A. Cas des graphes non orientés

Le problème de l'existence et de la détermination d'un cycle eulérien ou d'une chaîne eulérienne dans un graphe non orienté a été posé la première fois et résolu par Euler en 1736 à propos du célèbre problème des ponts de Königsberg. Euler prouva l'impossibilité de l'obtention d'une solution en démontrant le théorème suivant :

Théorème 1

Un graphe G = (X, U) contient un cycle eulérien si et seulement s'il est connexe et s'il n'y a aucun sommet de degré impair.

Un graphe G = (X, U) contient une chaîne eulérienne si et seulement s'il est connexe et si le nombre de sommets de degré impair est égal à 0 ou à 2.

Preuve

Le conditionnel est nécessaire : si un graphe contient une chaîne eulérienne, tout sommet du graphe, sauf le sommet de départ et le sommet d'arrivée s'ils sont distincts, est de degré pair. En effet, chaque passage sur un sommet du graphe utilise exactement deux arêtes. Pour prouver que le conditionnel est suffisant, on va montrer par l'absurde que si un graphe n'a que des sommets de degré pair, alors il a un cycle eulérien. Le cas où il y a exactement deux sommets de degré impair il faut y ajouter une arête fictive entre ces deux sommets de degré impair. On prend le plus grand (au sens du nombre d'arêtes) cycle C de G qui soit tel que $d_C(x)$ soit pair pour tout $x \in X$. Si C ne contient pas toutes les arêtes, alors il y a une arête $e = \{x, y\}$ qui a un point en commun avec C. En effet, si ce n'était pas le cas, C serait une composante connexe de G, ce qui est impossible car *G* est connexe. On suppose, sans perte de généralité, que *x* est dans C.

On construit une chaîne dans le graphe G – U(C) débutant en *x* avec cette arête, en utilisant chaque arête au plus une fois aussi longtemps que cela est possible de manière gloutonne. En tout sommet z ≠ x, il y a un nombre pair d'arêtes dans G – U(C). Par conséquent, la chaîne ne peut que revenir au bout d'un moment en *x*. En enchaînant ensuite avec les arêtes de C, on obtient un cycle eulérien plus long. D'où, la contradiction.

Corollaire 1

Un graphe non orienté G (X, U) est eulérien si et seulement si :

(i) $U \neq \varnothing$;

(ii) G est connexe, et

(iii) Tous les sommets sont de degré pair.

B. Cas des graphes orientés

Dans une ville où il y a des sens interdits ou sens uniques (comme dans la ville de Kananga : les avenues Machar et Lulua ainsi que les boulevards Lumumba), la question de savoir s'il est possible de parcourir toutes les rues en passant par chacune d'elles une fois et une seule est une question d'existence de parcours eulériens. Pour les parcours eulériens dans un graphe orienté, on a le théorème suivant, qui implique que la vérification de l'existence d'un parcours eulérien est simple (et polynomiale).

Théorème 2

Un graphe orienté G = (X, U) contient un circuit eulérien si et seulement s'il est connexe et si, pour tout sommet $x, d^+(x) = d^-(x)$.

Un graphe orienté G = (X, U) contient un chemin eulérien si et seulement s'il est connexe et si pour tout sommet $x, d^+(x) = d^-(x)$, sauf peut-être pour deux sommets s et t tels que $d^+(s) = 1 + d^-(s)$ et $d^-(t) = 1 + d^+(t)$.

La démonstration de ce théorème est similaire à celle du théorème 1.

Corollaire 2

Un graphe orienté G (X, U) est eulérien si et seulement si :

(i) $U \neq \varnothing$;

(ii) G est connexe, et

(iii) Pour tout sommet, le degré entrant est égal au degré sortant.

Algorithmes de construction d'un cycle ou circuit eulérien

Cas des graphes non orientés

Pour construire un cycle eulérien dans un graphe connexe G = (X, U) ayant tous ces sommets de degré pair, on peut appliquer l'algorithme simple de Fleury qui peut se décrire de la manière suivante :

Algorithme de Fleury

Début

1. Choisir un sommet x_0 et poser $C := \emptyset$;

2. Pour i allant de 1 à $|U|$:
   - Choisir une arête e_i adjacente à x_{i-1} qui ne déconnecte pas G – U(C) ;
   - Faire $C := C \cup \{e_i\}$ et définir x_i comme étant l'extrémité de e_i différente de x_{i-1}

Fin

Exemple :

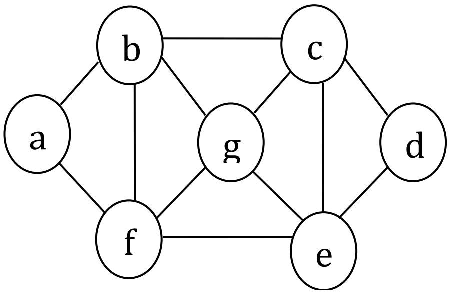


Fig. 2 *: Circuits eulériens : abcdefbgcegfa ou afgcdegbcefba*

Nota

- Pour montrer que cet algorithme est correct, il suffit de vérifier que l'on peut toujours choisir une telle arête e_i ce qui est évident car G – U(C) est connexe et possède d'après le théorème 1 une chaîne eulérienne qui débute en x_{i-1} (de degré impair dans G – U(C)).
- Pour trouver une chaîne eulérienne dans un graphe ayant 2 sommets de degré impair, il suffit d'ajouter une arête fictive entre les deux sommets de degré impair, appliquer l'algorithme de Fleury, puis supprimer l'arête fictive du cycle eulérien obtenu.
- Au lieu d'ajouter une arête entre les deux sommets de degré impair, on peut prendre pour x_0 l'un des deux sommets de degré impair.

Cas des graphes orientés

Pour savoir si un graphe orienté G(X, U) contient un circuit eulérien, il suffit de regarder tour à tour tous les sommets. S'il y a deux sommets s et t tels que :

$d^+(s) = 1 + d^-(s)$ et $d^-(t) = 1 + d^+(t)$, alors tout chemin eulérien a comme début s et comme fin t.

Pour construire un circuit dans un graphe orienté connexe G = (X, U) tel que tout $x \in X$, $d^+(x) = d^-(x)$, graphe appelé alors graphe eulérien, on peut appliquer une version adaptée de l'algorithme de Fleury.

Algorithme de Fleury bis

Début

1. Choisir un sommet x_0 et poser $C := \emptyset$;
2. Pour i allant de 1 à $|U|$:
   - Choisir une arête u_i quittant x_{i-1} qui ne déconnecte pas G – U(C) ;
   - Faire $C := C \cup \{u_i\}$ et définir x_i comme étant l'extrémité de $u_i \neq x_{i-1}$

Fin

Nota : Pour trouver un chemin eulérien dans un graphe ayant deux sommets s et t comme dans le théorème 2, il suffit de prendre $x_0 = s$ dans l'algorithme de Fleury bis.

Pour trouver un chemin eulérien, on applique l'algorithme d'Euler suivant :

1) Compter le nombre de sommets de degré impair,
2) Si
- C'est 0 : construire un cycle partant du premier sommet,
- C'est 2 : construire un chemin entre les deux sommets en question,
- Sinon, il n'y a pas de chemin eulérien.
3) Parcourir le chemin déjà construit : à chaque sommet
- Tant qu'il reste des arêtes partant de ce sommet dans le graphe,
- Construire un cycle partant de ce sommet et l'insérer dans le chemin.

Nota : Etant donné les conditions sur le graphe, construire un chemin ou un cycle n'est pas compliqué : il suffit de partir de l'origine, de prendre la première arête rencontrée et de recommencer récursivement. Se convaincre que c'est vrai, puis écrire la fonction qui :

- Renvoie un chemin entre deux sommets de degré impair dans un graphe eulérien
- Renvoie un cycle partant d'un sommet de degré pair dans un graphe eulérien dont tous les sommets sont de degré pair.

Les chemins et les cycles étant renvoyés sous forme de listes chaînées, la fonction devra supprimer les arêtes utilisées dans le graphe. Nous avons essayé d'expliquer la nature du problème eulérien tel que ce problème est relié à l'origine même de la théorie de graphe remontant à Euler. Nous avons constaté que c'est à partir de la notion de graphe orienté qu'on a la possibilité de solutionner le problème qui consiste à savoir si on peut parcourir toutes les rues d'une ville en passant par chacune d'elles une fois et une seule. Cette éventualité se vérifie ainsi dans le cas du problème du postier chinois.

Le problème du postier chinois

De quelle façon le parcours du postier chinois s'est-il réalisé à partir du procédé du graphe orienté, en tant que modèle du parcours eulérien ? Tel est le problème qui nous préoccupe sous ce point. Le postier n'est rien d'autre que le facteur : un agent qui passe de porte à porte pour déposer des lettres à leurs destinations. Mais comme les routes préexistantes ne sont pas tracées en fonction du passage du facteur, ce dernier sera-t-il en mesure de terminer tous ses parcours en n'empruntant chaque route qu'une seule fois ? Par ailleurs, les destinations étant multiples, et le point de départ étant connu, le facteur ne serait-il pas en train de suivre un graphe dont les différents destinataires seraient des sommets ? Partant ainsi de l'étymologie, nous arrivons à une définition scientifique disant que « le problème du postier chinois, problème du postier ou *Route inspection problem*, consiste à trouver un plus court chemin dans un graphe connexe non orienté qui passe seulement une fois par chaque arête du graphe et revient à son point de départ » **(33)**. Mais pourquoi l'épithète « chinois » ?

Le nom du problème vient du fait qu'il a été étudié par le mathématicien chinois Meigu Guan en 1962, et qu'il modélise le passage d'un facteur devant effectuer le plus efficacement possible sa tournée en passant une seule fois par chaque rue de son secteur **(31)**. C'est ceci qui constitue la différence principale avec le parcours eulérien. Le problème du postier chinois consiste à

British Journal of Environmental Sciences
Vol.11, No.2, pp.,46-66, 2023
Print ISSN: 2055-0219(Print)
Online ISSN: 2055-0227(online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development–UK

déterminer une tournée de coût total minimum qui passe une et une seule fois par un sous-ensemble fixé d'arêtes. En pratique, le réseau d'une ville est constitué de nombreuses arêtes et le ramassage des ordures ne doit se faire que sur certaines arêtes du graphe. Le camion chargé d'effectuer le ramassage peut cependant emprunter les autres arêtes du graphe pour se rendre d'un point à un autre.

Définition : supposons qu'une distance soit associée à chaque arête d'un graphe connexe G et soit D un sous-ensemble d'arêtes à desservir dans G. Le « problème du postier rural » est de déterminer un cycle aussi court que possible dans G qui passe une seule fois par chaque arête de D. Le problème du postier rural est très difficile à résoudre, comme celui du voyageur de commerce. En fait les problèmes réels sont bien plus complexes que ceux mentionnés ci- dessus puisqu'il faut tenir compte de nombreuses contraintes, tel que la capacité des véhicules, les pauses des chauffeurs, etc. Par exemple, lorsque la quantité totale à livrer ou à collecter auprès des clients dépasse la capacité d'un véhicule, il faut utiliser plusieurs camions. Le problème se complique donc puisqu'en plus de déterminer la route de chaque véhicule, il faut également répartir les clients parmi les véhicules.

Au-delà des tournées du facteur, le problème peut s'avérer très utile pour tout type de livraison (ou de collecte). Dans les grandes villes, que ce soit pour le nettoyage des rues ou pour les diverses livraisons commerciales, disposer d'itinéraires intelligents est un atout particulièrement intéressant en termes de coûts et de main d'œuvre.

Enoncé du problème

Etant donné un graphe $G = (X, U)$ et une fonction de *poids* $w : U \longrightarrow N^*$, trouver un cycle de poids minimum qui passe par toutes les arêtes au moins une fois.

Résolution du problème

La résolution du problème du postier revient à choisir un sous-ensemble de poids minimum $F \subseteq U$ d'arêtes tel qu'en doublant (c'est-à-dire introduisant une copie parallèle supplémentaire des arêtes de F) on obtient un graphe eulérien.

En effet, si on a un parcours de postier, alors en introduisant autant de copies parallèles de chaque arête que le nombre d'utilisations de l'arête par le postier, on obtient évidemment un graphe eulérien.

Inversement, en introduisant des copies parallèles de façon eulérienne, chaque cycle eulérien du graphe obtenu déterminera un parcours de postier de la même longueur (égale au poids total des arêtes ajoutées).

Enfin, il suffit de remarquer que, dans un parcours minimal, il n'y aura pas plus d'une copie supplémentaire d'une arête, puisque sinon on pourrait supprimer deux copies supplémentaires sans affecter ni la parité des sommets, ni la connexité, mais ceci diminuerait le poids total.

On a donc le problème suivant :

British Journal of Environmental Sciences
Vol.11, No.2, pp.,46-66, 2023
Print ISSN: 2055-0219(Print)
Online ISSN: 2055-0227(online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development–UK

Etant donné un graphe G = (X, U) et une fonction de poids $w : U \to \mathbb{N}^*$, trouver $F \subseteq U$ de poids W(F) minimum tel que pour tout sommet $x \in X$, son degré dans (X, F) ait la même parité que dans G (il faut noter que $W(F) := \sum_{u \in F} w(u)$).

Pour trouver $F \subseteq U$, nous faisons recours à la résolution d'un autre problème plus général celui de trouver un T-joint de poids minimum. Avant cela introduisons d'abord la notion de T-joint.

Définition d'un T-joint

Soient un graphe connexe G = (X, U) et $T \subseteq X$ tel $|T|$ pair. Un T-joint est un sous-ensemble $F \subseteq U$ tel que $d_F(x)$ est impair pour $x \in T$ et pair pour $x \notin T$

Problème du T-joint de poids minimum [2]

Etant donné un graphe G, une fonction de poids w : U → Z, un sous-ensemble $T \subseteq X$ de cardinalité paire, trouver un T-joint de poids minimum.

Nota

– Le problème du postier chinois est équivalent au cas particulier du problème du T-joint poids minimum, où T est l'ensemble de sommets de degré impair de G sachant que dans un graphe connexe l'ensemble des sommets de degré impair est de cardinalité paire.

– Rappelons la notion de différence symétrique entre deux ensembles A et B : $A \Delta B = (A \cup B) - (A \cap B)$

Lemme 1

Soit G = (X, U) un graphe connexe. Si F et F' sont respectivement des T- et T'- joints, alors F Δ F' est un TΔT'-joint.

Preuve

On a :

$d_{F \Delta F'}(x) \equiv \left(d_F(x) + d_F(x)\right)$ mod 2.

Donc si $x \in$ TΔT'

alors $d_{F \Delta F'}(x) \equiv 1$ mod 2

et si $x \notin$ TΔT'

alors $d_{F \Delta F}(x) \equiv 0$ mod 2.

Nota

– Si G = (X, U) est un graphe connexe et w une fonction de poids sur les arêtes, on note : $U\_ := \{u \in R : w(u)\}$ et $T\_ := \{x \in X : d_{\underline{U}}(x) \text{ est impair}\}$

– Si w : U → Z et $F \subseteq U$, on note w[F] la fonction w[F] : U → Z définie par w[F](u) = w(F) si $u \notin F$ et w[F](u) = -w(F) si u ∈ F où $w(F) := \sum_{u \in F} w(u)$

– Si w : U → Z, on note $|w|$ la fonction $|w| : U \to \mathbb{N}$ pour laquelle $|w|(u) := |w(u)|$ pour $u \in U$.

On a alors les propositions suivantes :

Proposition 1

Soit G = (X, U) un graphe connexe, w : U → Z une fonction de poids et T_0, $T \subseteq X$. Supposons que F_0, $F \subseteq U$ soient des T_0-joints et T-joints respectivement. Alors F est un T-joint de poids minimum avec des poids w si et seulement si FΔF_0 est un TΔT_0-joint de poids minimum avec des poids $w[F_0]$.

Preuve

Soit F_0 un T_0-joint et F un T-joint. D'après le Lemme 1, FΔF_0 est un TΔT_0-joint. L'égalité $w(F) = w[F_0](F\Delta F_0) + w(F_0)$ implique que w(F) est minimum si et seulement si $w[F_0](F\Delta F_0)$ est minimum.

Proposition 2

Soit G = (X, U) un graphe connexe, w une fonction de poids sur les arêtes et $T \subseteq X$. Alors F_+ est un TΔT_--joint minimum pour la fonction de poids w.

Preuve : Appliquons la proposition précédente à $F_0 := U_-$, $T_0 := T_-$ et $F := F_+ \Delta U_-$

Maintenant que nous avons vu que l'on peut toujours se ramener au cas où tous les poids sont positifs ou nuls, il nous reste à présenter l'algorithme du T-joint de poids minimum dans ce dernier cas.

Algorithme du T-joint de poids minimum dans le cas où tous les poids sont positifs ou nuls

Début

1. Pout toute paire de sommets $s, t \in T$ calculer la plus courte chaîne entre s et t ; noter : $\lambda(s,t)$ la longueur de la chaîne la plus courte.
2. Déterminer un couplage parfait M de poids min du graphe complet sur T avec les poids λ. Soient $s_i t_i (i = 1, \dots, |T|/2)$ les arêtes de ce couplage et $P_i (i = 1, \dots, |T|/2)$ les plus courtes chaînes correspondantes, c'est-à-dire $w(P_i) = \lambda(s_i, t_i)$.
3. Rendre $F := \bigcup_{i=1}^{n} P_i$

Fin

Théorème 3

L'ensemble F rendu par l'algorithme est un T-joint de poids minimum, et $w(F) = \sum_{i=1}^{n} (s_i, t_i)$.

Preuve : On montre d'abord que pour tout T-joint F' on peut construire un couplage de T avec un poids inférieur, donc min{w(F') : F' est un T-joint} ≥ valeur d'un couplage minimum pour λ. Si T = ∅, il n'y a rien à montrer. Si T ≠ ∅, on considère une composante de F' contenant un nombre non nul de sommets de T de degré impair dans cette composante. Cette composante doit en contenir un nombre pair. Soient $x, y \in T$ de deux sommets, et soit P une chaîne de F' qui joint x et y. Alors F' –P est un F – {x, y}-joint. Par récurrence, T admet un couplage parfait $\{s_i', t_i'\}$ $(i = 1, \dots, |T|/2 - 1)$ de poids inférieur à w'(F' – P) + λ(x, y) et donc inférieur à w'(F' – P) + w(P) = w(F').

Pour conclure, il suffit de montrer qu'à partir d'un couplage, on peut obtenir un T-joint F' de poids inférieur ou égal. Considérons donc un couplage de T, et les P_i correspondants.

Définissons $F' := \Delta_i^{|T|/2} P_i$. D'après le lemme 1, F' est un T-joint et l'on a bien$w(F) \leq \sum_{i=1}^{|T|/2} \lambda(s_i, t_i)$.

Par conséquent, pour trouver le parcours du postier chinois qui ait le poids le plus petit, il suffit de trouver le T-joint de poids minimum, de doubler ensuite les arêtes de ce T-joint, et enfin trouver un cycle eulérien dans ce nouveau graphe (par l'algorithme de Fleury).

Application au ramassage des ordures ménagères

En ce qui nous concerne, nous allons nous attarder sur le problème de *ramassage des ordures ménagères* qui inondent les rues et caniveaux de la ville de Kananga en appliquant le parcours eulérien selon le schéma du graphe orienté au modèle du postier chinois.

Dans la ville de Kananga, le principe de ramassage des ordures ménagères n'est pas tellement d'application, cet état de choses est essentiellement dû à un problème de politique locale en la matière. Cette absence de politique de salubrité dans la ville de Kananga laisse à la population la possibilité de gérer à sa manière des déchets ménagers, en les jetant tout simplement dans la rue, et en pratiquant ce que Binzangi appelle « poubellisation ». **(2)** La situation ainsi décrite apparaît comme une conséquence de la négligence et du laisser-aller de la part des autorités provinciales, locales et celles de nos quartiers qui devraient chercher par tous les moyens à réglementer la gestion des déchets et ordures ménagers depuis leur émission jusqu'à l'imposition des écotaxes. Ainsi sommes-nous intéressée à définir les concepts qui permettent de comprendre la situation des déchets.

Considération générale

Quelques concepts nécessitent d'être clarifiés avant de voir notre analyse s'appliquer à la situation de ramassage des immondices de ménage dans les rues de la ville de Kananga. Ces concepts seront définis en relation avec l'importance de la question de la protection de l'environnement.

Environnement

L'Environnement est perçu comme un cadre de vie, un réservoir des ressources et un réceptacle des déchets. *L'Environnement* naturel renvoie généralement à la masse des êtres et des espèces qui entourent l'univers vivant.

Déchets

On appelle déchet tout résidu d'un processus de production, de transformation ou d'utilisation, toute substance, matériaux, produit ou, plus généralement, tout bien meuble abandonné ou que son détenteur destine à l'abandon. Par leur nature, on distingue les déchets solides, les déchets liquides qui sont pompables et les déchets gazeux, c'est-à-dire toute substance gazeuse libérée par l'incinération d'un objet ou par une réaction chimique quelconque qui puisse conduire à une nuisance ou pollution (résidus de bois, rebuts de caoutchouc, fumées des différentes industries).

De ce qui précède, nous retenons que nous avons différents types de déchets : des déchets agricoles, industriels, ménagers et assimilés. Dans la présente étude, nous nous intéressons spécialement au type de déchets ménagers. Ces déchets ménagers sont des déchets dégradables de l'alimentation, associés à d'autres qui sont non dangereux.

Fig. 4 : Abandon sauvage des immondices dans les rues

Le ramassage est le fait de collecter les déchets afin de les sortir devant la maison avant de les faire stocker dans le bac à ordures public. Ce bac sera vidangé par vélo-chariot, (pour le cas de Kananga où il n'existe pas d'autres moyens de transport plus appropriés), qui ira les déposer à un lieu aménagé et autorisé. Malheureusement pour le cas qui nous concerne, ces déchets sont abandonnés dans toutes les artères, les bifurcations et voire devant les parcelles.

Fig. 5 : Bacs à ramassage des immondices.

Nuisance

La nuisance se dit de la dégradation d'une situation, suite à une modification négative de l'environnement, par une action produisant des perturbations à conséquences écologiques pathologiques de nature à porter atteinte à l'intégrité de la vie en général.

Pollution

La pollution, c'est l'action de salir en rendant malsain et dangereux l'environnement habitable.

Assainissement

L'assainissement est défini comme l'action visant à l'amélioration de toutes les conditions qui, dans le milieu physique de la vie humaine, influent ou sont susceptibles d'influer défavorablement sur le bien-être physique, mental ou moral, la santé et la longévité de cette vie. Après la définition de ces concepts, il convient de vérifier en quels termes se pose le problème de gestion des déchets dans la ville de Kananga par la modalité de l'évacuation dans la poubelle publique à partir du parcours relevant du modèle eulérien.

Modélisation et Implémentation

Modélisation

Le chemin le plus court sur le réseau routier est-il possible ?

Une représentation schématique l'illustrera : soit G un graphe fictif d'ordre 8, un réseau routier. Partant d'un point donné **A**, quel serait le chemin le plus court pour atteindre les autres sommets, en l'occurrence **H** ?

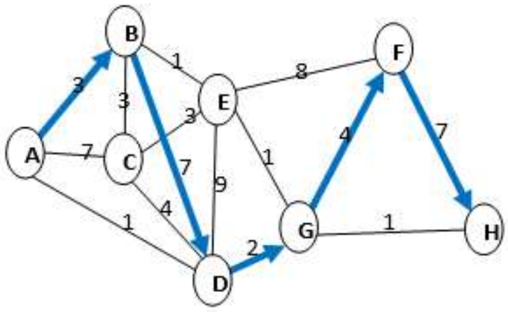


Fig. 6 : Le chemin le plus court allant d'un sommet à un autre.

Pour parcourir tous les sommets en partant de **A**, le chemin le plus court serait :

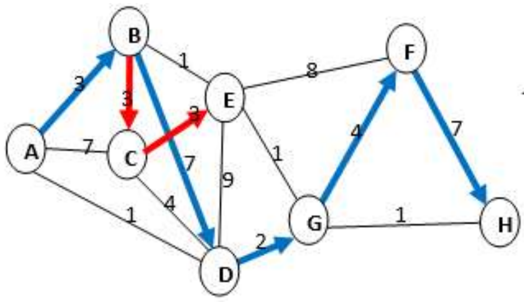


Fig. 7 : Le chemin le plus court allant d'un sommet à tous les autres.

Quelle que soit la méthode, Dijstra arrive aux solutions ci-haut. Mais ce n'est qu'une solution partielle, car certaines artères restent non parcourues : B-E ; E-G ; E-F ; E-D ; A-D ; A-C ; C-D ; G-H. La question eulérienne est beaucoup plus englobante, car elle vise le parcours de toutes les artères en ou en minimisant le coût.

Retour sur le parcours eulérien

Le terme « parcours » est plus large que le terme « cycle », car dans le parcours eulérien on trouve aussi bien le chemin que le cycle d'Euler. Par ailleurs, sachant que le chemin fait état de la succession d'arcs, ou plus généralement, une suite contigüe d'arêtes, la spécificité eulérienne tient

British Journal of Environmental Sciences
Vol.11, No.2, pp.,46-66, 2023
Print ISSN: 2055-0219(Print)
Online ISSN: 2055-0227(online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development–UK

au fait que le passage par chaque arête se fait une et une seule fois. La figure ci-dessous en témoigne :

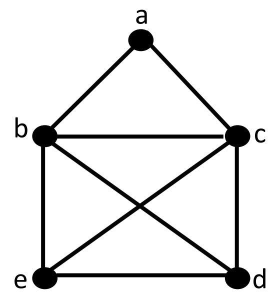


Fig.8 : Le chemin eulérien : e - b - d - e - c - a - b - c – d
Toutes les huit arêtes sont parcourues, mais chacune en une et une seule fois.

Cela conduit à la définition du chemin eulérien comme « un chemin dans le graphe qui passe par toutes les arêtes juste une seule fois ». Au cas où ce chemin est fermé, on parle d'un cycle eulérien **(14)**. Laissant un moment de côté cette vision scientifique, nous pouvons revenir à un jeu enfantin qui consiste à tracer une figure d'un trait, sans soulever la plume et sans revenir sur la même ligne. Ceci explique le chemin trouvé sur ce graphe.

En gros, la notion de cycle eulérien entraîne celle du graphe eulérien : un graphe contenant un cycle eulérien. Mais à quelles conditions peut-on parler d'un circuit ou d'un chemin eulérien ? Deux théorèmes répondent à cette double question :

1°) « Un multi-graphe connexe admet un *cycle eulérien* si et seulement si chacun de ses sommets est de degré pair » **(32)** C'est « la condition d'unicursalité » **(27)**.
Ex : Soit G le graphe non orienté d'ordre 5.

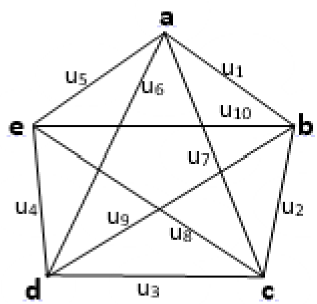


Fig. 9 : Un graphe complet d'ordre 5 qui est en même temps un graphe eulérien.

$d_G(a) = d_G(b) = d_G(c) = d_G(d) = d_G(e) = 4$

Par le fait que le degré de chaque sommet est un nombre pair, ce graphe est d'une connexité qui permet de parcourir par une chaîne tous ses sommets. On aura des cycles comme : $C_{10}= (u_1, u_2, u_3, u_4, u_{10}, u_9, u_6, u_7, u_8, u_5)$

$C_{10}= (u_5, u_1, u_2, u_3, u_9, u_{10}, u_8, u_7, u_6, u_4)$

Un cycle de ce genre est appelé cycle eulérien et il répond au critère de traçage d'une figure sans soulever la plume et sans revenir sur des traces préexistantes. Ce théorème engendre des considérations telles que :

– Un graphe complet d'ordre impair supérieur à 1 est toujours et déjà un graphe eulérien.
– Un graphe régulier pair est eulérien.

Mais est-ce possible pour chaque cas ? Répondons tout de suite par la négative, car tous les graphes n'ont pas toujours des sommets pairs. Ce qui entraîne un autre théorème.

2°) « Un multi-graphe connexe admet un *chemin eulérien* si et seulement s'il a exactement deux sommets de degré impair » Dans ce cas l'un servira de point de départ et l'autre de point d'arrivée et vice versa.

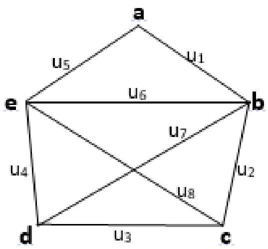


***Fig. 10** : L_8= (u_8, u_4, u_3, u_2, u_6, u_5, u_1, u_7)*
soit c – e – d – c – b – e – a – b – d
Parce d_G(c) et d_G(d) sont impairs, alors ce graphe contient un chemin commençant par c pour sortir par d en parcourant toutes les arêtes une seule fois chacune.

Il y a des cas qui ne répondent ni aux conditions de cycle eulérien ni à celles de chemin eulérien. Que faire ? Des solutions sont multiples, mais on va en épingler deux : la duplication des arêtes et la scission du multigraphe en plusieurs sous-graphes.

- *La duplication des arêtes*

Si l'objectif c'est la recherche du chemin optimal et que ce dernier, dans sa nature cumulative, impose de revenir 1+n fois sur la même arête, « il est plus simple de considérer l'approche alternative du problème : plutôt que de permettre d'emprunter plusieurs fois la même arête, on duplique les arêtes du graphe par lesquelles on souhaite passer plusieurs fois. Le but est alors de compléter le graphe pour le rendre eulérien, en minimisant la longueur totale des arêtes ajoutées. On obtient une solution du problème initial en cherchant un circuit eulérien dans le graphe complété. On rappelle pour la suite que dans toute composante connexe d'un graphe, la somme des degrés des sommets est le double du nombre d'arêtes, donc est paire » **(33)**.

- *La scission du multigraphe en plusieurs sous-graphes*

L'analyse de l'utilisateur du réseau symbolisant le multigraphe débouchera sur plusieurs sous-graphes répondant à l'un des théorèmes évoqués ci-haut. Cela nous conduit à une application possible sur le ramassage des ordures dans les rues de la ville de Kananga.

Ramassage des ordures sur les routes de la ville de Kananga

Le réseau routier de la ville de Kananga inclut toutes les cinq communes urbaines, à savoir : Kananga, Katoka, Ndesha, Lukonga et Nganza. La complexité d'un tel réseau est indubitable et cette raison nous pousse à cerner un échantillon pour démontrer notre théorie. Nous avons pensé à un multigraphe représentant les routes du centre-ville et nous tâcherons de retrouver, si possibles, des circuits et des chemins selon les indications d'Euler. A première vue, ni le circuit ni le chemin eulérien ne peuvent être possibles, car non seulement tous les sommets ne sont pas de degré pair, mais surtout les sommets à degré impair dépassent largement le nombre deux. Il faut alors tenter de dupliquer les routes ; tentative difficile et presqu'irréalisable pour notre province. La seule possibilité qui reste est celle de diviser le réseau routier en plusieurs sous-graphes.

Recherche des cycles eulériens

L'échantillon représentatif qui est ici permet de retrouver certains cycles pour le parcours des ramasseurs sans revenir sur une même route deux fois. Cela est possible si nous divisons le multigraphe en plusieurs sous-graphes quadrilatères à arêtes directement contigües. Exemple : B – C - G - H - B

On retiendra que le graphe eulérien contient autant de points de départ qu'il y a de sommets. Ainsi B – C – G - H - B peut devenir C – G – H - B - C

Cette méthode est bonne, mais elle crée une multiplicité inouïe de sous-graphes. Il faudrait envisager une autre voie.

3.2.1. Recherche des chemins eulériens

On les trouvera en tenant compte du théorème relatif à cet effet.

Ex : s – E - D – F –K – r – s – F (pour s et F de degré impair)

3.2.2. Implémentation

L'implémentation peut résoudre le problème annoncé avec plus d'efficience. Nous avons, à cet effet, utilisé le langage de $C^{\#}$ en mode console, en nous basant sur la cartographie de la ville de Kananga ci-jointe en annexe.

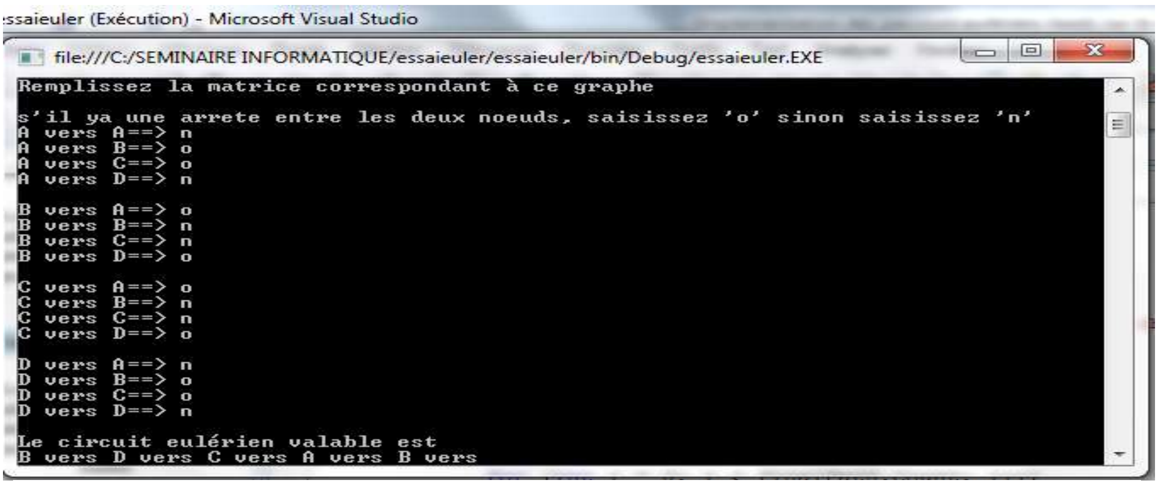
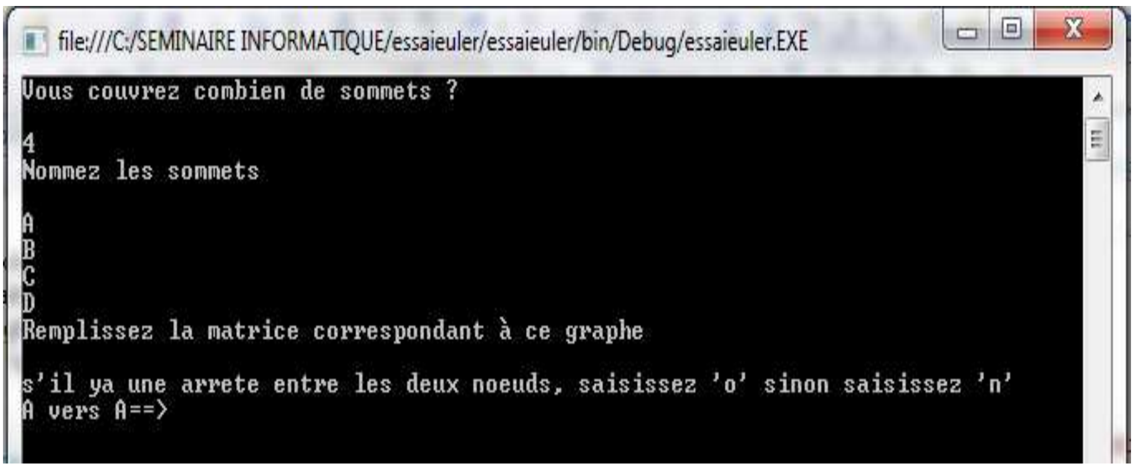


Fig.11 : Au lancement, un message vous demande le nombre de sommets du graphe visé. Une fois le nombre de sommets saisi, le programme demande de les nommer.



British Journal of Environmental Sciences
Vol.11, No.2, pp.,46-66, 2023
Print ISSN: 2055-0219(Print)
Online ISSN: 2055-0227(online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development–UK

Fig. 12 : Après avoir établi toutes les relations, le programme donne le circuit ou le chemin ou déclare qu'il est impossible de trouver un circuit ou un chemin.

Le problème du parcours des rues d'une ville en passant par chacune d'elles une fois et une seule est donc lié au problème d'existence d'un cycle eulérien dans le graphe où les sommets sont les intersections de rues, et les arêtes sont les parcours de rues entre les intersections.

Mais, il est évident que les rues de la ville de Kananga ne constituent pas toujours un graphe eulérien. La question qui se pose est de déterminer un parcours optimal permettant de passer au moins une fois dans chacune d'elles. C'est le problème du « postier chinois » ou « facteur chinois ». Pour résoudre ce problème on procède de la manière suivante : on considère que le graphe *G= (X, U)* représente les rues de la ville de Kananga. Il s'agit, dès lors, de déterminer un cycle de coût minimal passant au moins une fois par chaque arête de U. Dans pareil cas de figure, si G est eulérien, il suffit de construire un cycle eulérien par application de l'algorithme de Fleury. Si G n'est pas eulérien, on trouvera le T-Joint de poids minimum *(en utilisant l'algorithme du T-Joint de poids minimum)*, on double ensuite les arêtes de ce *T-Joint*, et enfin on trouve un cycle eulérien dans ce nouveau graphe par l'algorithme de Fleury.

Nous avons appliqué l'algorithme de Fleury pour le ramassage des immondices ménagères (en annexe son code source en *C# en mode console.* Il va ainsi de soi que si ces algorithmes sont traduits en programmes, la résolution du problème du ramasseur des ordures dans la ville de Kananga sera facilitée et accélérée, surtout si le graphe comporte de nombreux nœuds et arêtes, comme on le voit dans la cartographie de Kananga.

Codes Sources

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;
namespace essaieuler
{
    class EulerCircuit
    {
        Stack tempPath = new Stack();
        ArrayList finalPath = new ArrayList();
        char[] lessommets;
        char[,] lesrelations;
        int total, count;
        private void Saisie()
        {
            Console.WriteLine("Vous couvrez combien de sommets ? \n");
            try
            {
                total = int.Parse(Console.ReadLine());
                lesrelations = new char[total, total];
                lessommets = new char[total];
                Console.WriteLine("Nommez les sommets \n");
                for (int i = 0; i < total; i++)
                {
                    lessommets[i] = char.Parse(Console.ReadLine());
                }
    Console.WriteLine("Remplissez la matrice correspondant à ce graphe \n");
    Console.WriteLine("s\'il ya une arête entre les deux noeuds, saisissez 'o'
 sinon saisissez 'n'");
```

```
                for (int i = 0; i < total; i++)
                {
                    for (int j = 0; j < total; j++)
                    {
  Console.Write("{0} vers {1}==> ", lessommets[i], les sommets[j]);
       les relations[i, j] = char.Parse(Console.ReadLine());
                    }
                    Console.WriteLine("");
                }
            }
            catch
            {
                Console.WriteLine("Invalide");
            }
        }
       private int saisiedegre(int i)
        {             int j, deg = 0;
            for (j = 0; j < total; j++)
{
                if (les relations[i, j] == 'o') deg++;
            }
            return deg;
        }
        private int trouverracine()
        {
            int rac = 1;
            count = 0;
            for (int i = 0; i < total; i++)
            {
                if (saisiedegre(i) % 2 != 0)
                {
                    count++;
                    rac = i;
                }
            }
            if (count != 0 && count != 2)
            {
                return 0;
            }
            else return rac;
        }
        private int GetIndex(char c)
        {
            int index = 0;
            while (c != les sommets[index])
                index++;
            return index;
        }
        private Boolean AllVisited(int node)
        {
            for (int l = 0; l < total; l++)
            {
                if (les relations[node, l] == 'o')
                    return false;
            }
            return true;
        }
        private void FindEuler(int root)
        {
            int ind;
            tempPath.Clear();
            tempPath.Push(lessommets[root]);
            while (tempPath.Count != 0)
            {
                ind = GetIndex((char)tempPath.Peek());
                if (AllVisited(ind))
```

```
int saisiedegre(int i)
        {
                {
                    finalPath.Add(tempPath.Pop());
                }
                else
                {
                    for (int j = 0; j < total; j++)
                    {
                        if (les relations[ind, j] == 'o')
                        {
                            Les relations[ind, j] = 'n';
                            Les relations[j, ind] = 'n';
                            tempPath.Push(lessommets[j]);
                            break;
                        }
                   }
                }
            }
        public void lecircuit()
        {
              Saisie();
            int racine = trouverracine();
            if (racine != 0)
            {
                if (count != 0) Console.WriteLine("Le chemin eulérien valable est ");
                else Console.WriteLine("Le circuit eulérien valable est ");
                FindEuler(racine);
                PrintEulerCircuit();
            }
            else
            {
                Console.WriteLine("Impossible de trouver un chemin eulérien");
            }
        }
        public void PrintEulerCircuit()
        {
            for (int i = 0; i < finalPath.Count; i++)
            {
                Console.Write("{0} vers ", finalPath[i]);
            }
        }
    }
class ExecuteEulerCircuit
    {
        static void Main(string[] args)
        {
            EulerCircuit ec = new EulerCircuit();
            ec.lecircuit();
            Console.ReadKey();
        }
    }
}
```

REFERENCES

(1). Albertit, G., *Pas de visas pour les déchets : vers une solidarité Africaine/Europe en matière d'environnement,* éd. Harmattan, Paris, 1990.
(2). Augustin Muzumbi Mukamba, *assainissement urbain par l'approche « pollueur payeur » au quartier Matonge, dans la commune de Kalamu à Kinshasa*, IFAD, Kinshasa, inédit, 2008.
(3). Berge C., *Théorie des graphes et ses applications*, Collection Universitaire de Mathématiques, éd. Dunod, Paris, 1958 ;
(4). Bertolini G., *Déchet, mode d'emploi*, Economica, 1996 ;
(5). Bertolini G., *Le marché des ordures : économie et gestion des déchets ménagers*, éd. L'Harmattan, collection Environnement, Courtine, 1996 ;
(6). Bretto, A., Moukrim, A., *Eléments de théorie des graphes*, collection IRIS, 2012 ;
(7). Carlier et P Chrétienne, *Problèmes d'ordonnancement, modélisation, complexité, algorithme,* Masson, ISBN 2-225-81275-6
(8). Denegre J., et F. Salge, *Les systèmes d'information géographique*, éd. PUF, Paris, 2004 ;
(9). Didier Maquin, *Eléments de Théorie de graphes*, Version provisoire, mai 2003, Institut National Polytechnique de Loraine;
(10). Douglas B. West, *Introduction to graph theory,* 2001, printice-Hall;
(11). Ecoto, *Initiation à la recherche opérationnelle*, Ellipes, Paris, 2011 ;
(12). Edmonds J., and E.L. Johnson, Matching Euler, *tours and the chinese post man problem, mathematical programming*, 1973;
(13). Flament, C., *Théorie des graphes et structures sociales*, éd. Mouton, Paris, 1968 ;
(14). Frédéric Meunier et Andras Sebos, *Parcours et coupes*, Laboratoire Leibniz, mars 2006, site internet : http://www-leibniz.imag.fr
(15). Gondran M., Minoux M., *Graphes et Algorithmes*, 2ème éd. Eyrolles, 1986 ;
(16). Haray, F., N, R., et Cartwright, D., *Introduction à la théorie des graphes*, éd. Dunod, Paris, 1968 ;
(17). Jean Marc Nicod et Philippe Canalda, *Graphe et Optimisation*, Université de Paris, Comité 2004-2005
(18). Kondani, K., *Ethique et déchets*, Université de Kinshasa, Faculté de psychologie et des sciences de l'éducation, Kin, 1999 ;
(19). Maïmona Traole, *Analyse de la participation citoyenne autour de la gestion des déchets dans la ville d'Ouagadougou* (Bourkina Faso), 2011.
(20). Maïmona, Traole, Le *« sale » et le « propre »* : modes de gestion des déchets ménagers et logiques identitaires à Ouagadougou, Thèse de Doctorat en Anthropologie, Université de Poitiers 2011 ;
(21). Matula, A., *Pour une anthropologie des déchets-saleté, souillure et salubrité de nos cultures d'origine à Kinshasa,* Université de Kinshasa, Département de sociologie et anthropologie, Kin, 1999 ;
(22). Mavinga, N., *L'approche juridique de la gestion de déchets*, méd. Fac. : Landbouwn Univ. 64/1 in : actes du 1er colloque sur la problématique de déchets à Kinshasa ;
(23). MEUNIER F. et SEBO A., *Parcours et coupe*, Paris, Dunod, 2003.

(24). Mouhoub Nasser Eddine, *Algorithme de construction de graphes dans les problèmes d'ordonnancement de projet*, thèse de doctorat, Université Ferhat Abbas –Setif, 2011 ;
(25). N. Xuong (?), *Mathématique discrète et informatique*, Masson, Paris, 1991 ;
(26). NIKOLAJ VAN OMME, *Le problème du postier chinois cumulatif. Thèse présentée en vue de l'obtention du diplôme de Philosophiae Doctor*, Université de Montréal, Mai 2011.
(27). PERRIER N., *Problème du postier chinois mixte avec pénalité pour les virages. Mémoire présenté en vue de l'obtention du Grade de maître ès sciences (M.Sc.)*, Montréal, 2001.
(28). Reinchard Diestel, *Graph theory*, Springer verlag, Heidelberg, New York, 1997, 2000, 2005. Version Electronique de la 3ème edition;
(29). REY A., *Le Robert micro. Dictionnaire de la langue française,* Maury-Eurolivres, Paris, 2006.
(30). Roux Ch., *Initiation à la théorie des graphes,* éd. Ellipses, 2009 ;
(31). WARUSFEL A., *Euler : les mathématiques et la vie*, Vuibert, 2009.
(32). YENDE R. Grevisse et KASEKA K. Viviane « *Divergence possible des processus de Data mining et Knowledge Discovery in Databases* », European Journal of Natural and Social Sciences, EJNSS-NOVUS, 01(10), Janvier 2023
(33). YENDE R. Grevisse, TSHIELA K. Marie-Alice et al, *"Possible dangers of electromagnetic waves from cell phones and relay antennas on human health in DRC: Effects, Risks, Health and Protection*", European Journal of Mechanical Engineering Research, 10 (1),26-45, March 2023