Université Paris Cité

Postier chinois (non-orienté) et voyageur de commerce

CM nº5 — Mobilité (M2 IMPAIRS)

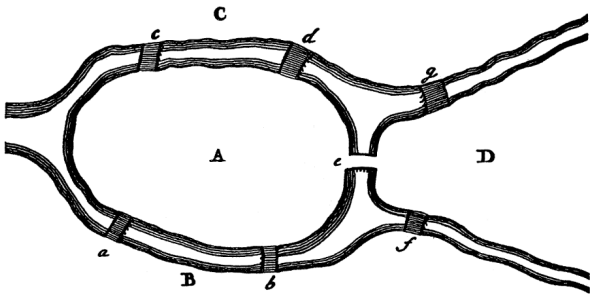
Matěj Stehlík

9/2/2024

Les sept ponts de Königsberg (18ème siècle)

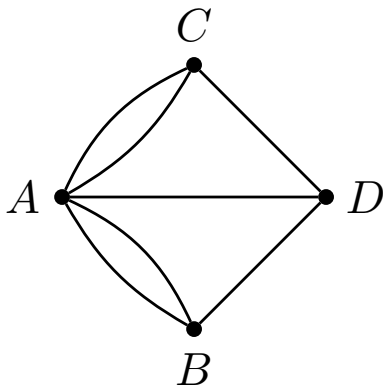
Problème

Existe-t-il 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 ?



Une reformulation

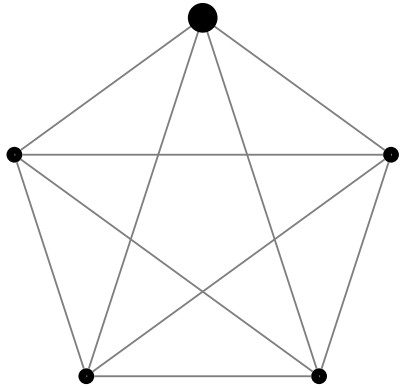
Existe-t-il un cycle dans le graphe à droite qui traverse chaque arête exactement une fois ?



Graphes eulériens

Définition

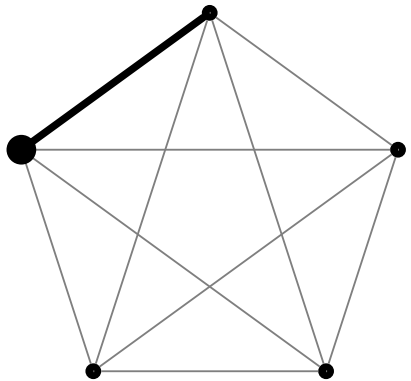
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

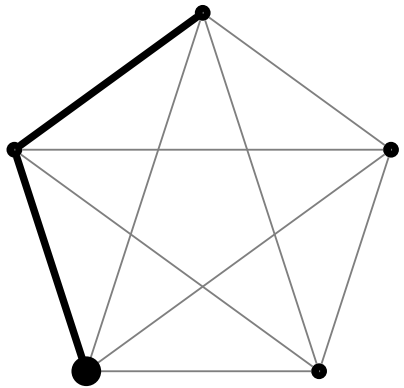
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

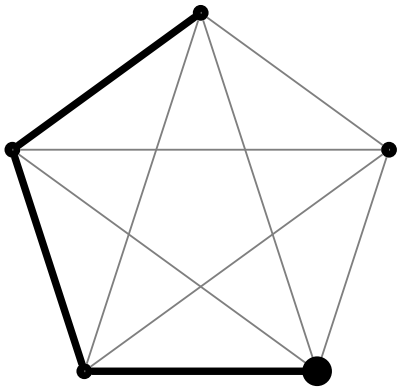
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

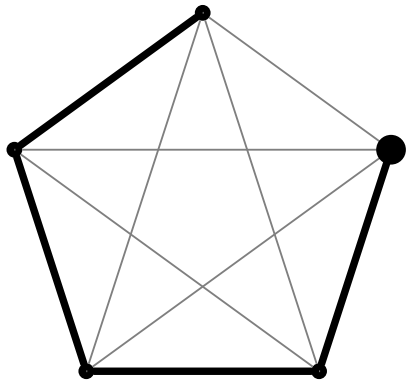
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

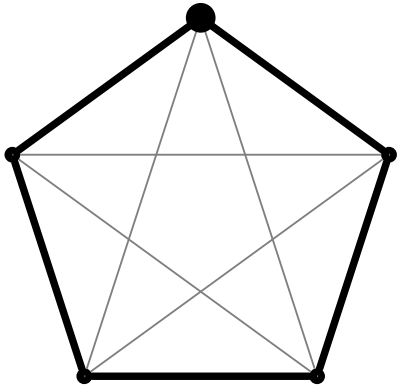
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

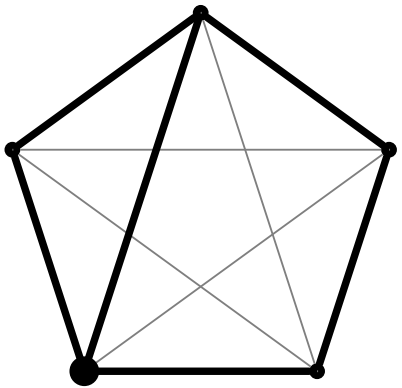
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

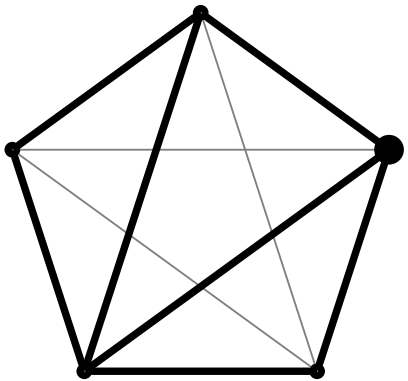
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

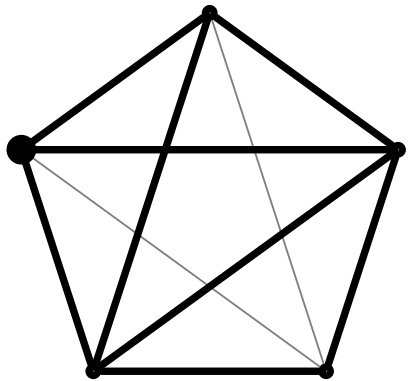
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

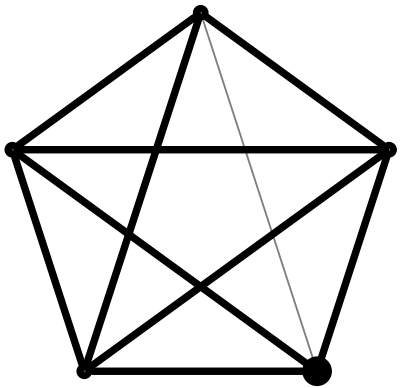
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

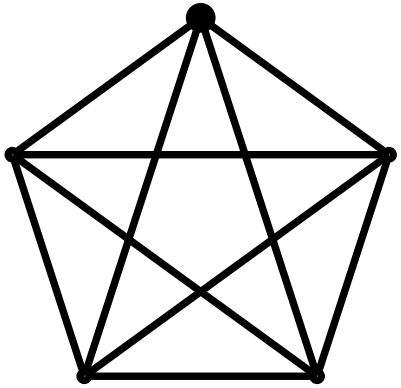
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



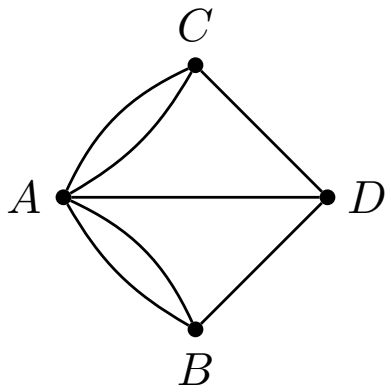
Caractérisation des graphes eulériens

Théorème (Euler 1736)

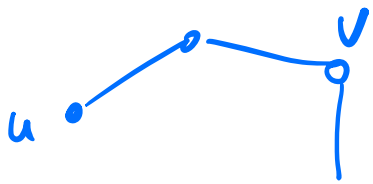
Un graphe G est eulérien si et seulement si G est connexe, et tout sommet de G est de degré pair.

Remarques

- Euler a prouvé eulérien $\implies$ connexe et tous les degrés pairs
- Il a ainsi donné une réponse négative au problème des sept ponts.

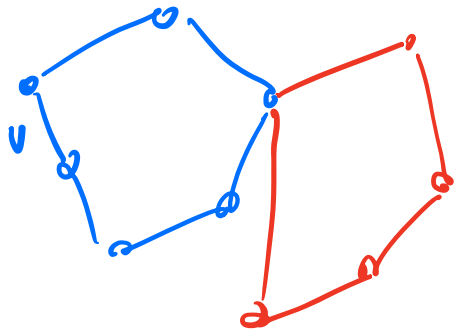


Démonstration du théorème d'Euler



- ($\Longrightarrow$)
- Soit G un graphe eulérien avec un cycle eulérien C avec sommet initial et terminal u.
- Chaque fois que C passe par un sommet $v \neq u$, on comptabilise deux arêtes incidentes à v.
- Donc, le degré de v est pair.
- De même, $d(u)$ est pair puisque C débute et termine en u.
- ($\Longleftarrow$)
- Conséquence de l'algorithme de Hierholzer.

L'algorithme de Hierholzer



- Choisir n'importe quel sommet initial v
- Suivre un chemin arbitraire d'arêtes jusqu'à retourner à v, obtenant ainsi un cycle C.
- Tant quil y a des sommets u dans le cycle C avec des arêtes qu'on n'a pas encore choisies faire
  - Suivre un chemin à partir de u, n'utilisant que des arêtes pas encore choisies, jusqu'à retourner à u, obtenant un cycle C
  - Prolonger le cycle C par C'

Justification de l'algorithme de Hierholzer

- Du fait de la parité du degré des sommets, si on entre dans un sommet, on peut toujours en sortir.
- La recherche du premier cycle finit tôt ou tard par terminer puisqu'on finit nécessairement par retourner dans le sommet initial v : l'ensemble des arêtes est fini et le sommet v a un nombre impair d'arêtes inexplorées une fois qu'on en est parti (donc il reste au moins une arête pour y entrer).
- Ces arguments sont encore vrai pour la recherche du cycle autour de u, ce qui fait que l'algorithme de Hierholzer termine.
- Il produit bien un cycle qui visite toutes les arêtes du graphe, sinon, par connexité, au moins un des sommets devraient avoir encore une arête non visitée.
- Finalement, il ne peut pas visiter deux fois la même arête, par construction : il produit donc bien un cycle eulérien.

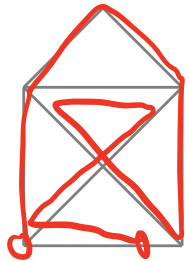
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

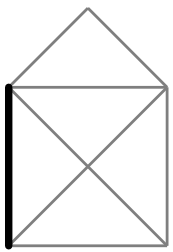
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

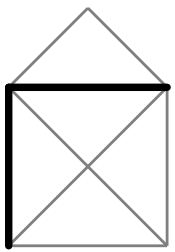
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

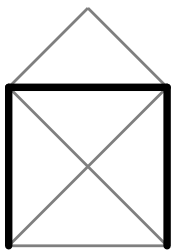
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

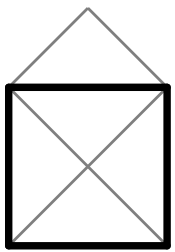
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

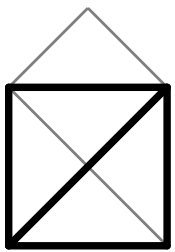
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

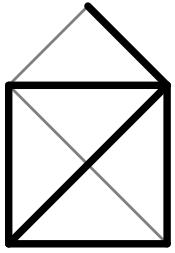
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

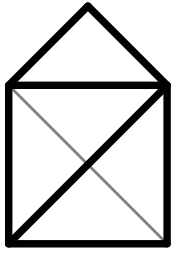
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

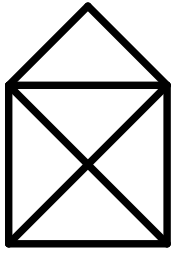
Chaînes eulériennes

Définition

Soit G un graphe et u, v deux sommets distincts de G. Une chaîne de u à v qui traverse chaque arête de G une et une seule fois est une chaîne *eulérienne*.

Théorème

Un graphe G contient une chaîne eulérienne de u à v ssi G est connexe et le degré de tous les sommets sauf u et v est pair.



Application

Les problèmes du type « dessiner sans lever le crayon ».

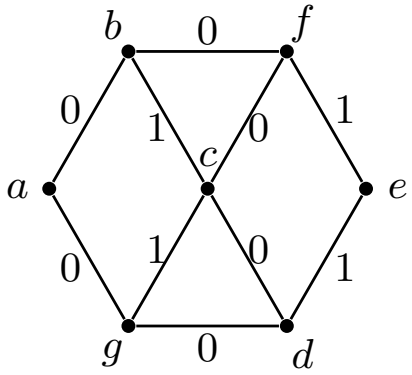
Démonstration

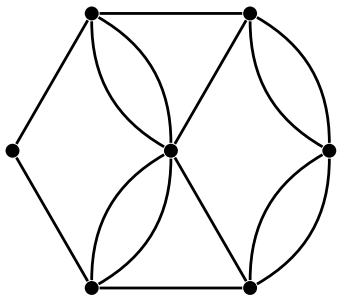
Démonstration

- ($\Longrightarrow$)
- Même argument que pour les cycles eulériens.
- ($\Longleftarrow$)
- Si l'on ajoute l'arête uv à G, on obtient un graphe connexe avec tous les sommets de degré pair.
- Par le théorème d'Euler, il existe un cycle eulérien C dans $G + uv$.
- Si l'on supprime l'arête uv de C, on obtient une chaîne eulérienne de u à v.

Le problème du postier chinois (version non orientée)

- Le même argument qu'on a utilisé dans le cadre des graphe orientés marche aussi pour les graphes non-orienté.
- Le théorème d'Euler peut être utilisé plus généralement.
- Supposons qu'on a un tour de postier chinois dans G.
- Soit x_e le nombre de traversées supplémentaires de l'arête e, pour toute $e \in E$.
- Soit G^x le supergraphe de G avec $1 + x_e$ copies de l'arête e, pour toute $e \in E$.





Le problème du postier chinois

- Il est facile à voir que G^x a un cycle eulérien.
- Supposons que $x \in \mathbb{Z}^E$ est non négatif, avec la propriété que G^x ait un cycle eulérien.
- Puis il existe un tour de postier chinois de G qui emprunte chaque arête e $1 + x_e$ fois.
- Au vu de ces observations et le théorème d'Euler, on peut reformuler le problème de la façon suivante :

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e & & \\ \text{s.c.} \quad & x(\delta(v)) \equiv d(v) \pmod 2 & & \text{pour tout } v \in V \\ & x_e \geq 0 & & \text{pour toute } e \in E \\ & x_e \in \mathbb{N} & & \end{aligned}$$

Reformulation du problème du postier chinois

- Il y a une solution optimale où x prend les valeurs dans $\{0, 1\}$: si $x_e \geq 2$, on peut diminuer x_e de deux, ce qui donne une autre solution de coût inférieur ou égal à celui de la solution originale.
- Il suffit donc chercher un ensemble J d'arêtes (avec vecteur caractéristique x).
- J est un *ensemble de postier* de G si pour tout sommet $v \in V$, v est incident avec un nombre impair d'arêtes dans J ssi v est de degré impair dans G.
- Le problème peut être reformulé comme suit :

Problème du postier chinois

Entrée Un graphe $G = (V, E)$ avec pondération $c \in \mathbb{R}^{|E|}$ tel que $c \geq 0$.

Objectif Trouver un ensemble postier J tel que $c(J)$ soit minimum.

Résolution du problème du postier chinois

- Soit $d(u, v)$ le coût d'une (u, v)-chaîne de coût minimum dans G.
- Soit $|T| = 2k$.
- Le coût minimum d'un ensemble de postier est :

$$\begin{aligned} \min \quad & \sum_{i=1}^{k} d(u_i, v_i) \\ \text{s.c.} \quad & u_1v_1, \ldots, u_kv_k \text{ est un jumelage (couplage) des éléments de } T. \end{aligned}$$

- C'est un problème de couplages !
- On forme un graphe complet $\hat{G} = (T, \hat{E})$, on affecte le poids $d(u, v)$ à chaque arête uv, et on trouve un couplage parfait de poids minimum.
- Ceci détermine un jumelage des sommets de T, et donc un ensemble de chaînes qui rejoignent les sommets jumeaux.

Algorithme de postier chinois (version non-orientée)

Entrée : Graphe $G = (V, E)$ avec une pondération $c \in \mathbb{R}^{|E|}$ t.q. $c \geq 0$.

Sortie : Un ensemble de postier $J \subseteq E$.

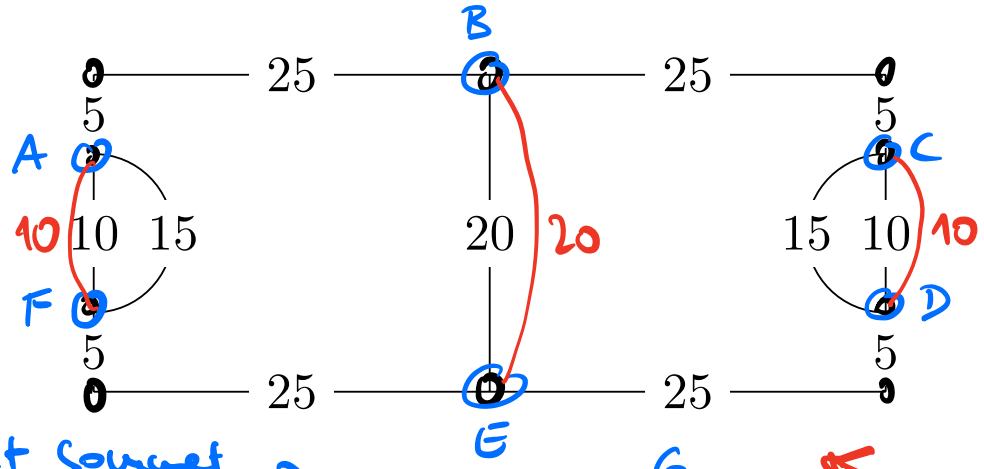
1. Pour chaque paire $\{u, v\}$ de sommets, trouver un (u, v)-chemin P_{uv} de coût minimum. Soit $d(u, v)$ le coût (longueur) de P_{uv}.
2. Soit $\hat{G} = (T, \hat{E})$ un graphe complet où l'arête uv est de poids $d(u, v)$, pour toute arête $uv \in \hat{E}$. Trouver un couplage parfait M de poids minimum dans G.
3. Soit J la différence symétrique des arêtes des chemins P_{uv} pour $uv \in M$.

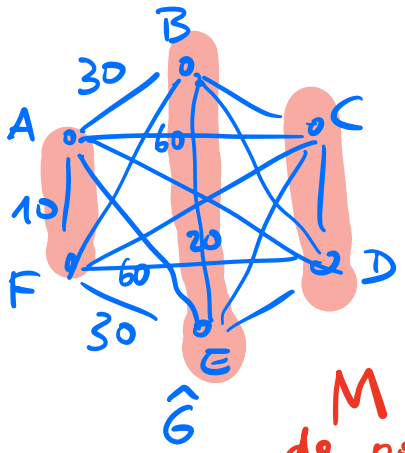
Exemple

Le jardinier doit marquer les lignes blanches sur un terrain de foot à 5, dont les distances (en mètres) sont données dans le diagramme suivant.

Couplage : Un ensemble M d'arêtes t.q. $e \cap f = \emptyset$ pour toute paire $e, f \in M$ t.q. $e \neq f$.

Couplage parfait : tout sommet est dans une arête





M de poids 40

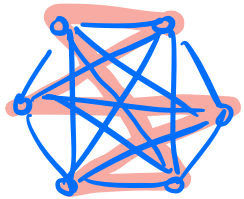
L'appareil à marquer les lignes étant cassé, il ne peut pas être arrêté et marque des lignes partout où il passe.

Question

Quelle distance doit parcourir le jardinier ?

C'est la somme de tous les poids

Le probleme du voyageur de commerce (TSP)



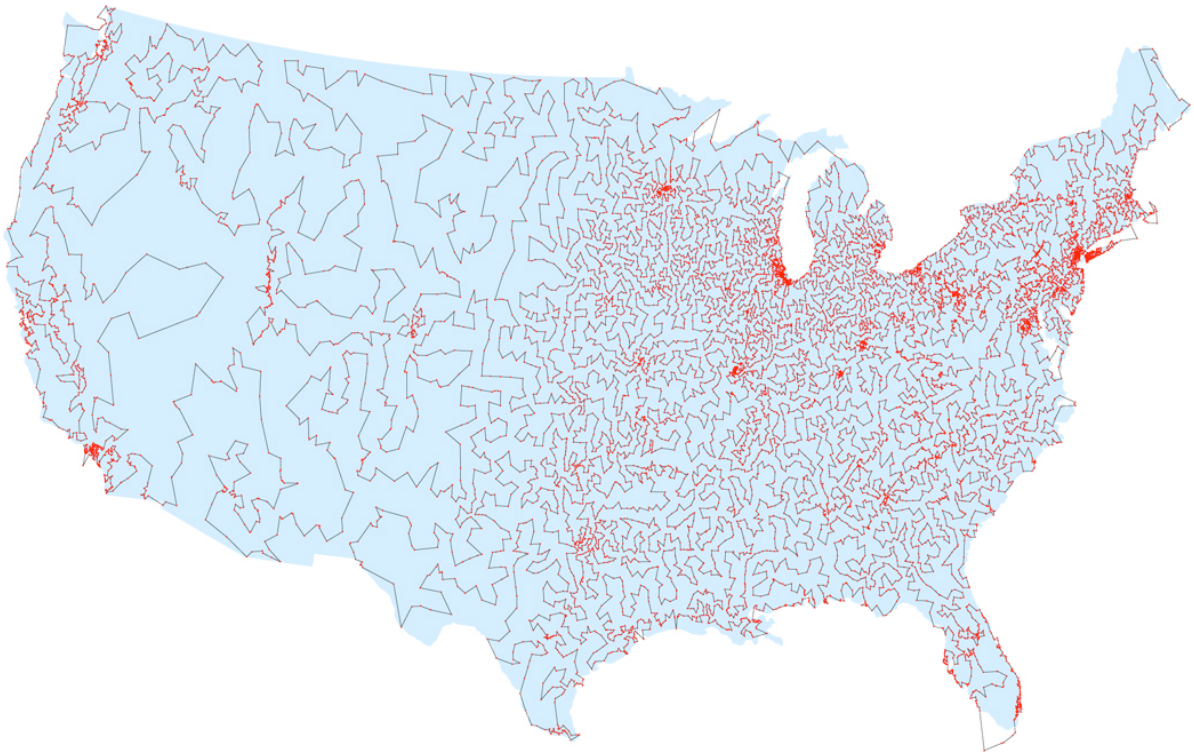
Le problème du voyageur de commerce (TSP)

Instance Un graphe complet $G = (V, E)$ avec pondération $w \in \mathbb{R}^{|E|}$.

Problème Trouver un cycle de poids minimum qui passe par chaque sommet de G une et une seule fois.

- Illustration dans la vie réelle : déterminer, étant donné une liste de villes et les distances entre toutes les paires de villes, le plus court cycle qui passe par chaque ville une et une seule fois.

Exemple d'un tour de voyageur de commerce aux États-Unis



TSP est NP-difficile

- Décider s'il existe un TSP de longueur au plus ℓ est un problème NP-difficile.
- Supposons qu'il y a un algorithme polynomial A pour résoudre Decision-TSP.
- Soit G un graphe quelconque.
- Ajouter des arêtes à G pour obtenir un graphe complet H t.q. $V(G) = V(H)$.
- arêtes originelles : poids 0, nouvelles arêtes : poids 1
- G contient un cycle hamiltonien ssi H contient un TSP de longueur 0.
- Donc, on peut utiliser A pour résoudre Hamiltonian Cycle (qui est NP-difficile) en temps polynomial.

L'heuristique Nearest Neighbor pour TSP euclidéen

Un cycle hamiltonien est construit de la façon suivante :

1. Commencer par un sommet quelconque
2. Trouver un sommet pas encore visité le plus proche au dernier sommet visité, et ajouter l'arête entre ces deux sommets
3. Répéter jusqu'à ce que le dernier sommet a été visité ; ajouter l'arête entre ce dernier sommet et le sommet initial.

Exemple de l'heuristique Nearest Neighbour

Exemple de l'heuristique Nearest Neighbour

Exemple de l'heuristique Nearest Neighbour

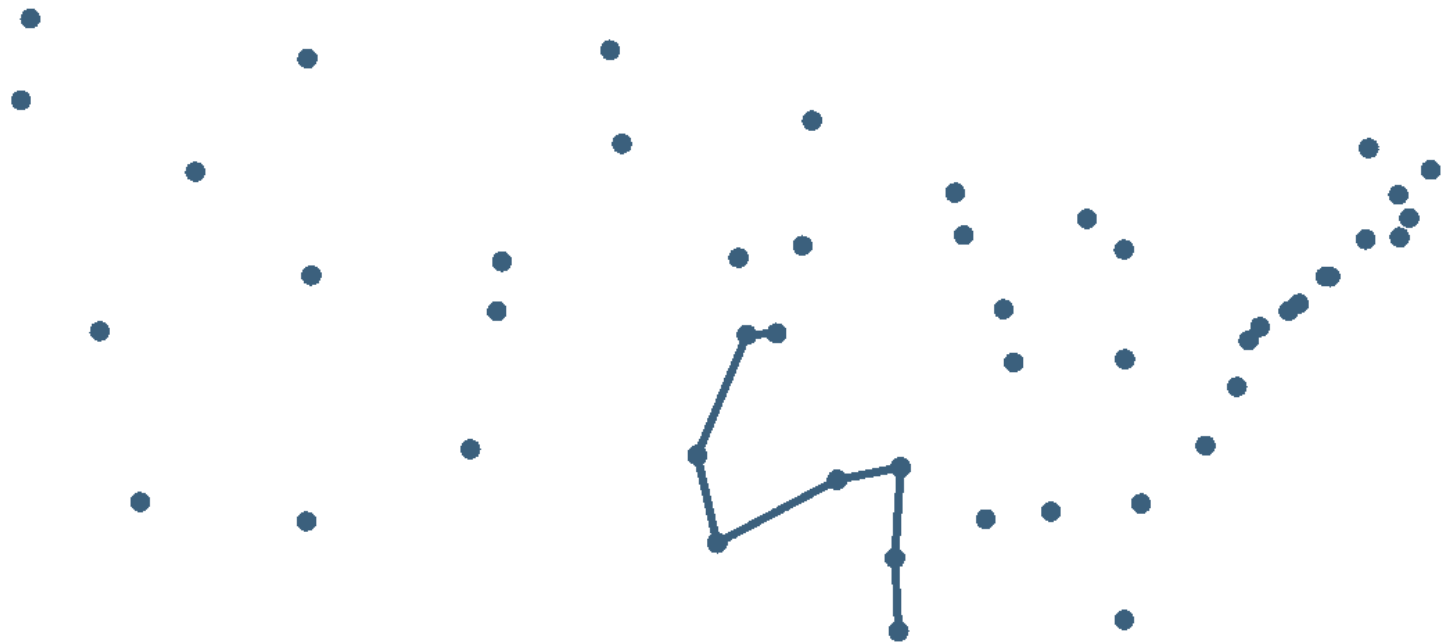
Exemple de l'heuristique Nearest Neighbour

Exemple de l'heuristique Nearest Neighbour

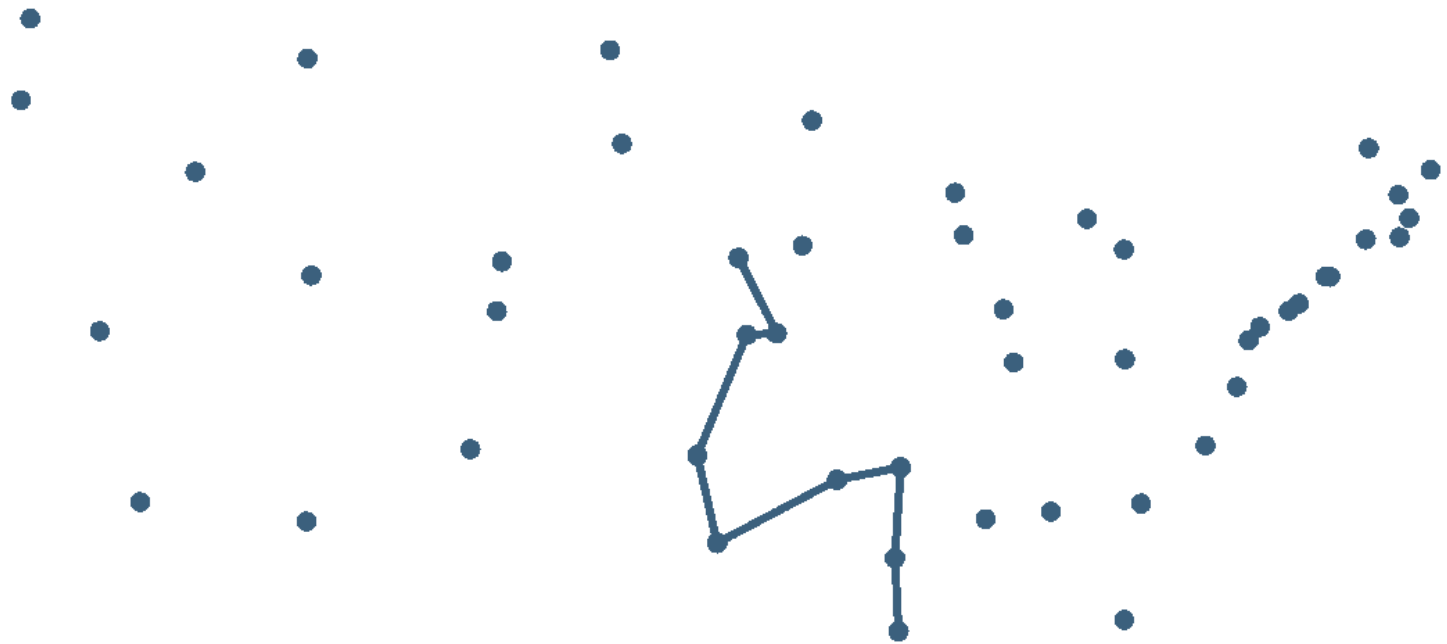
Exemple de l'heuristique Nearest Neighbour

Exemple de l'heuristique Nearest Neighbour

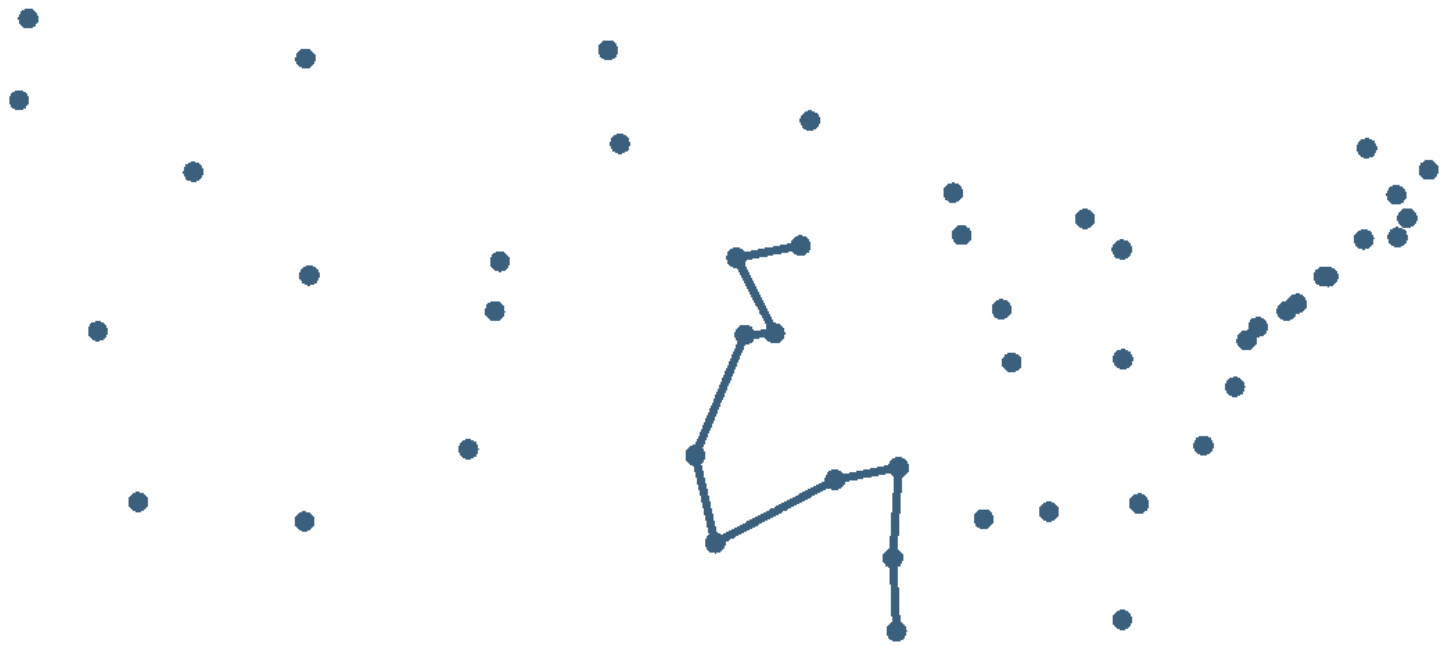
Exemple de l'heuristique Nearest Neighbour



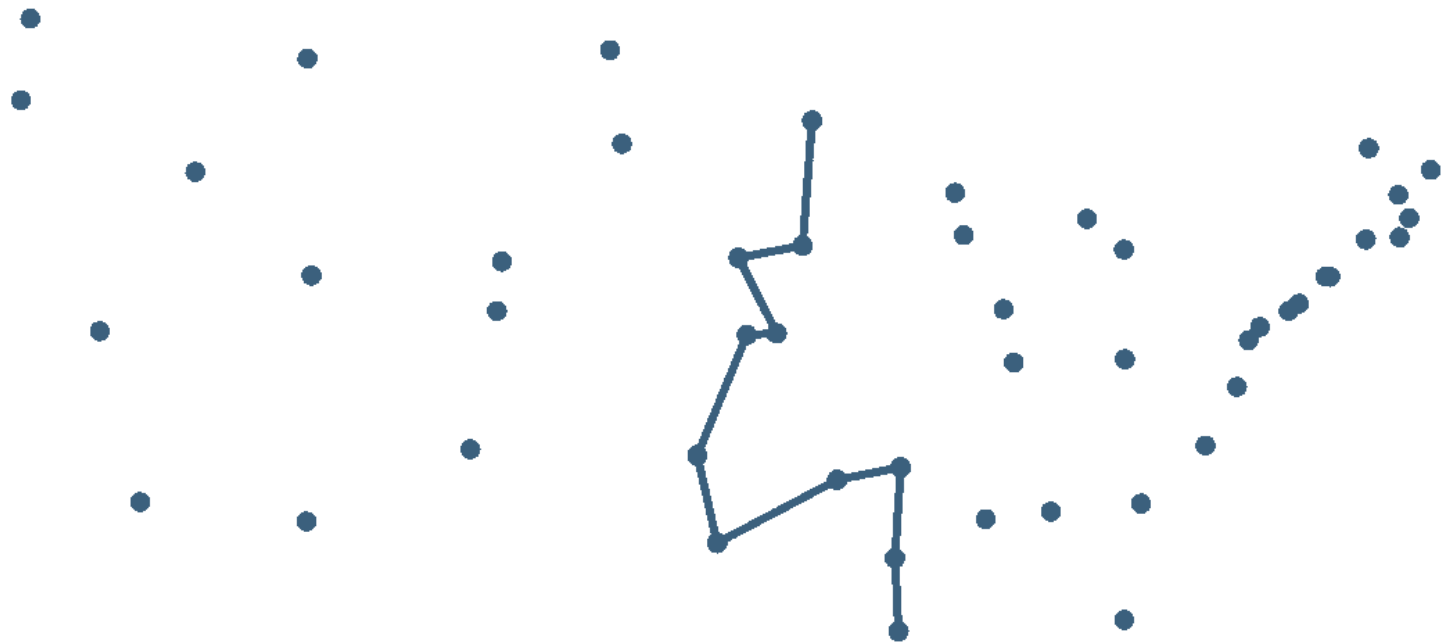
Exemple de l'heuristique Nearest Neighbour



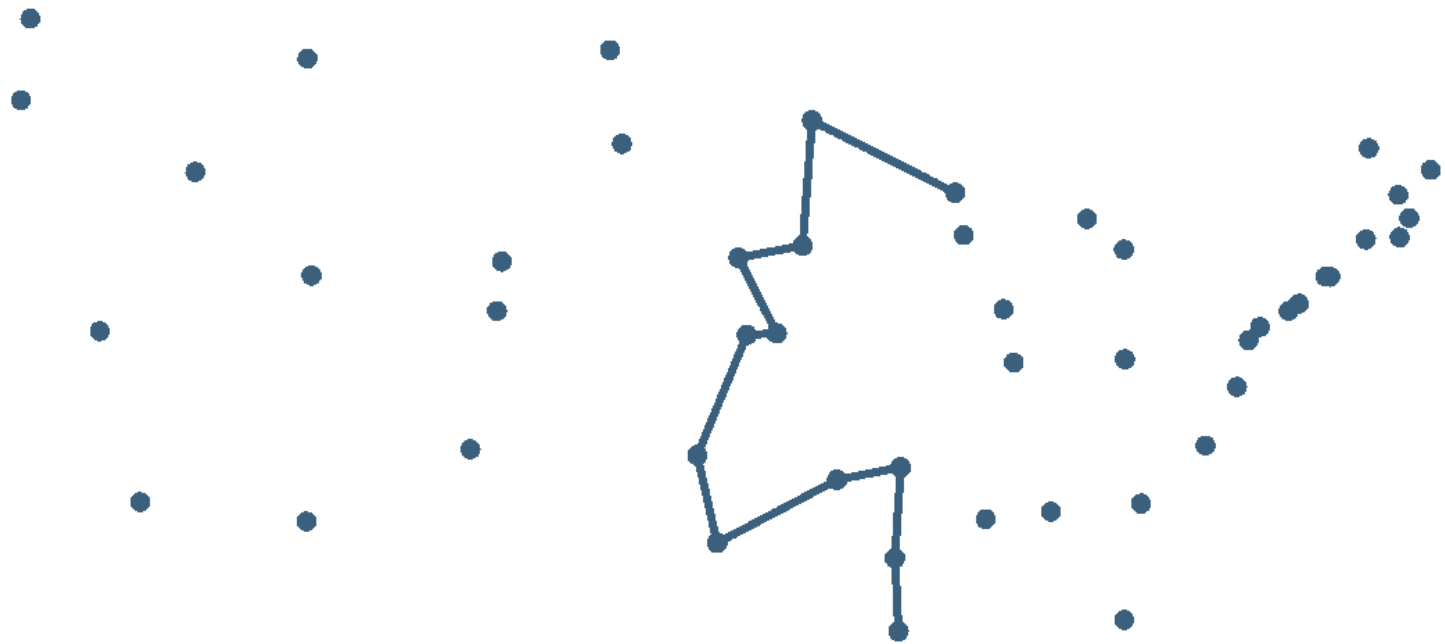
Exemple de l'heuristique Nearest Neighbour



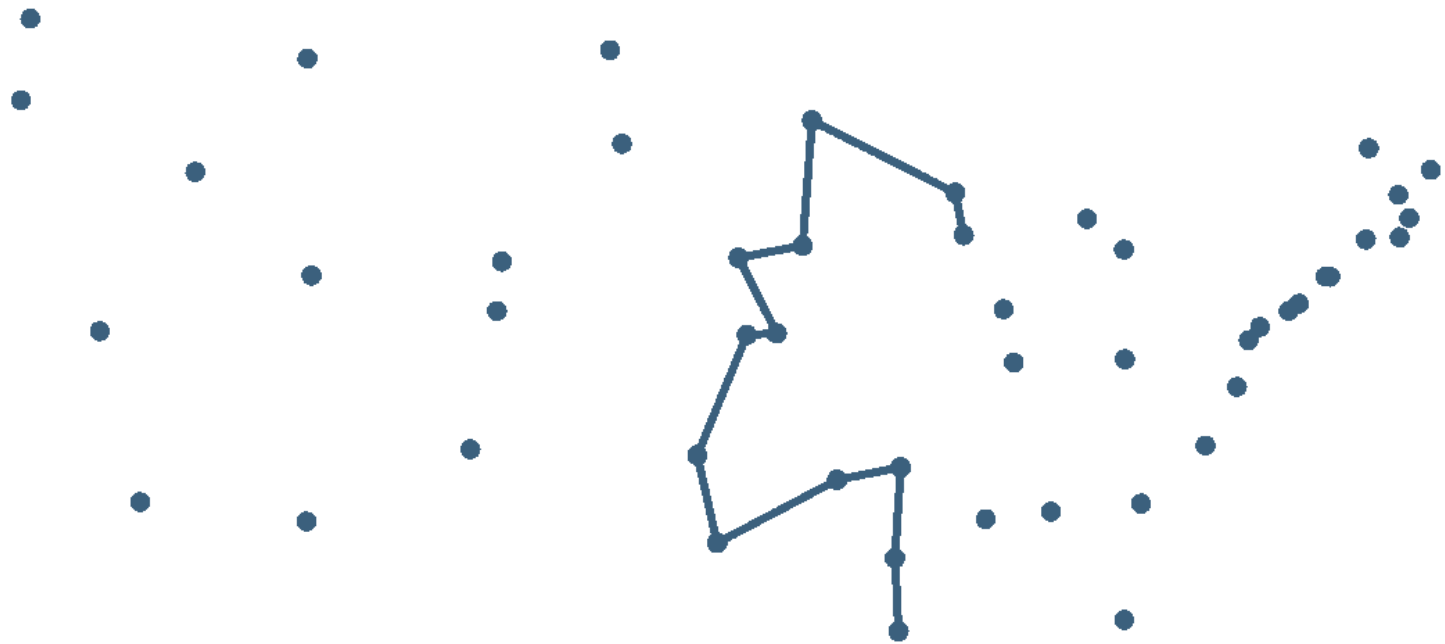
Exemple de l'heuristique Nearest Neighbour



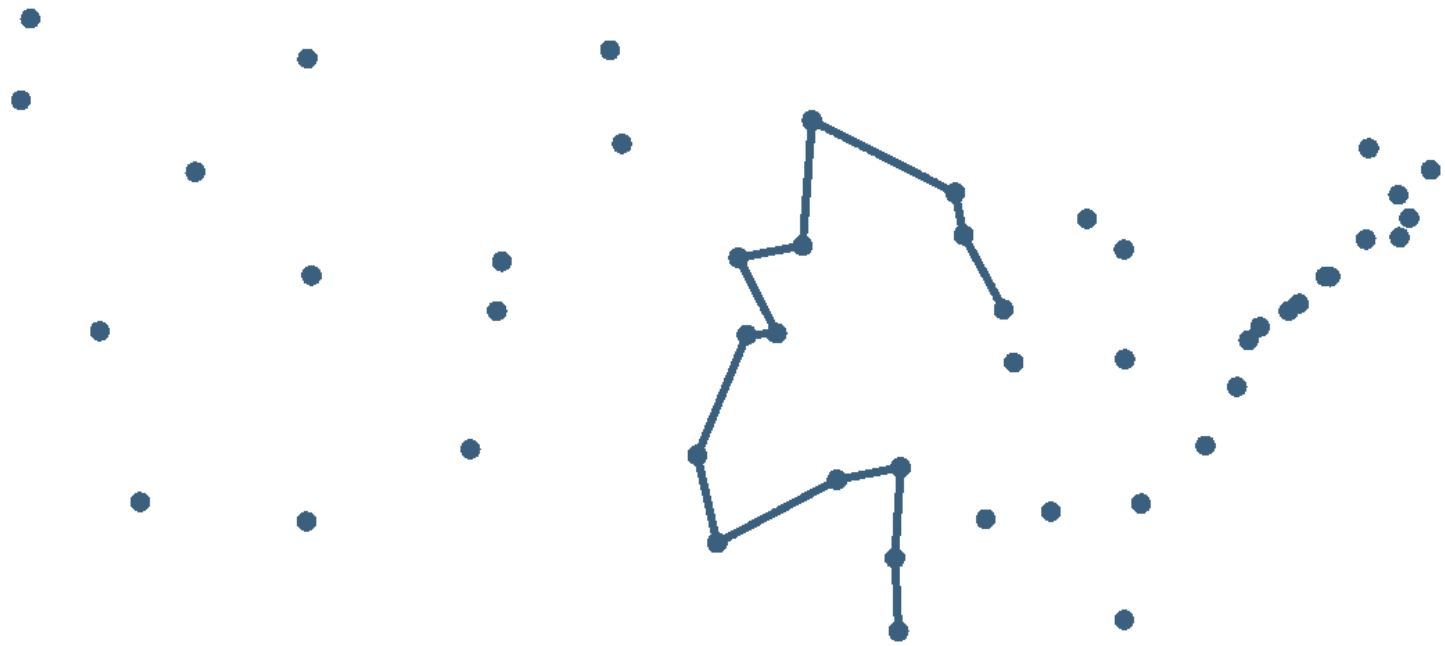
Exemple de l'heuristique Nearest Neighbour



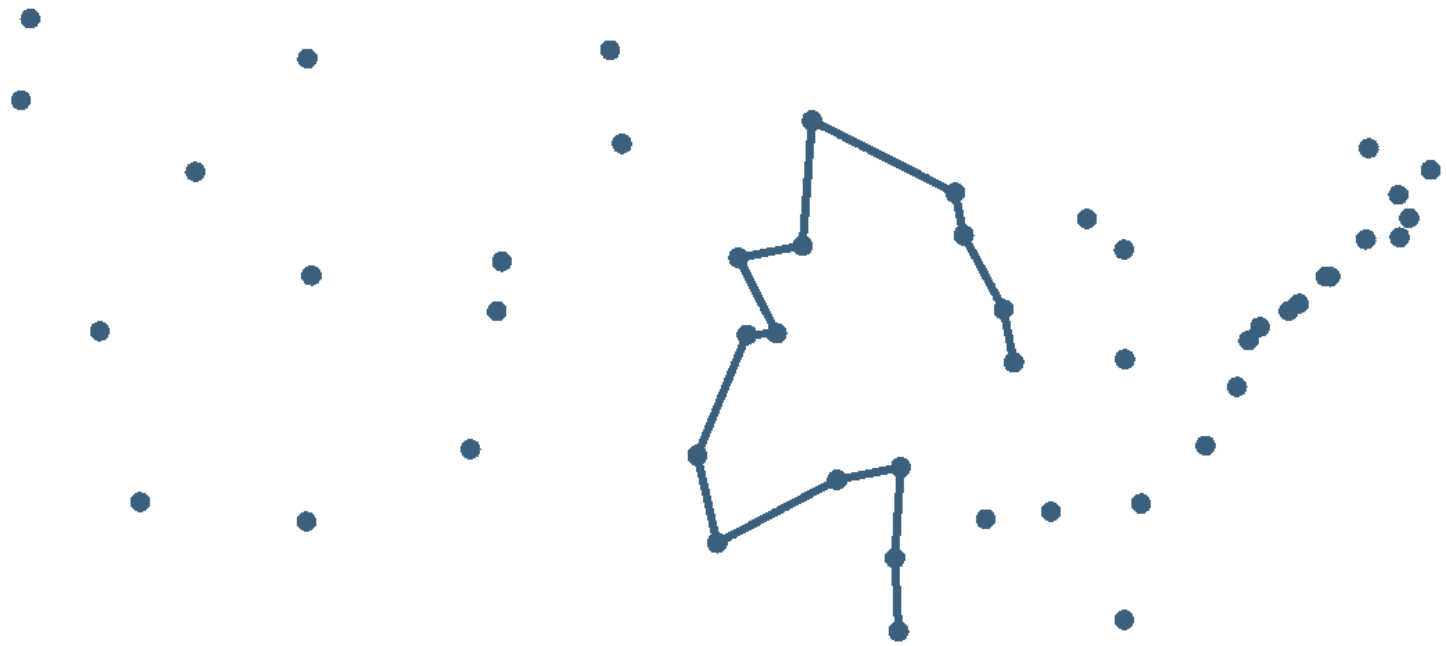
Exemple de l'heuristique Nearest Neighbour



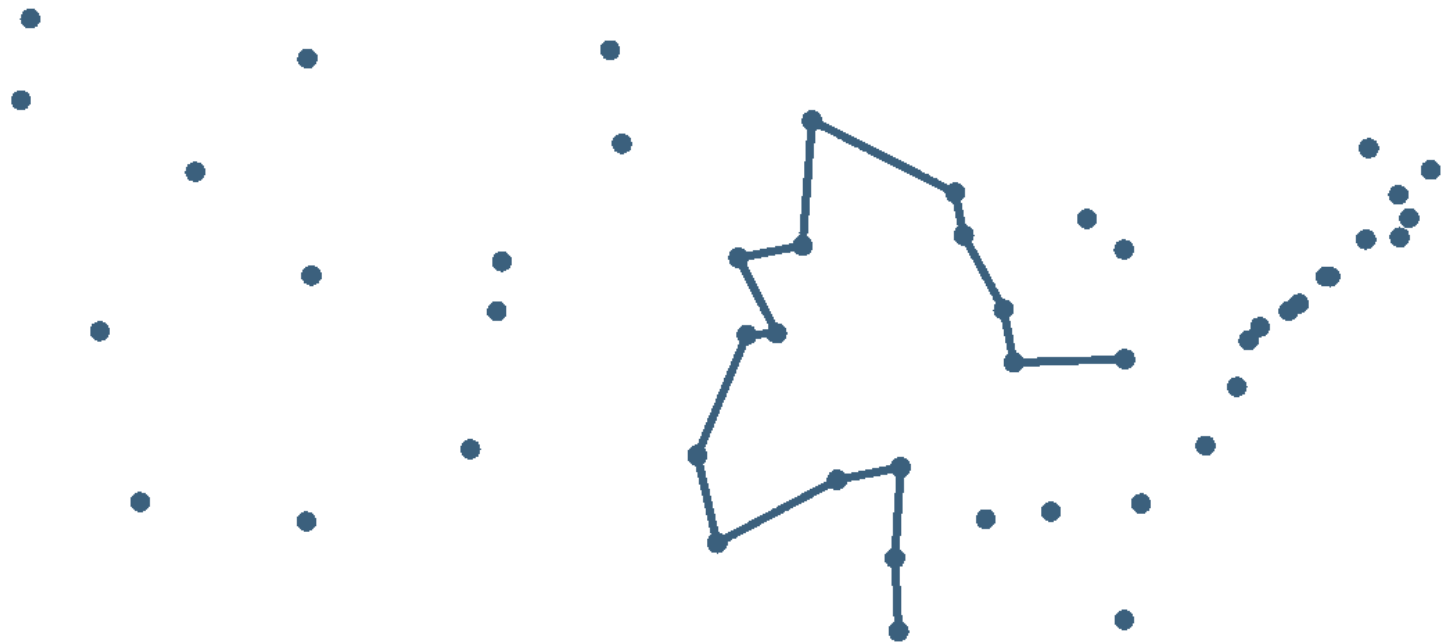
Exemple de l'heuristique Nearest Neighbour



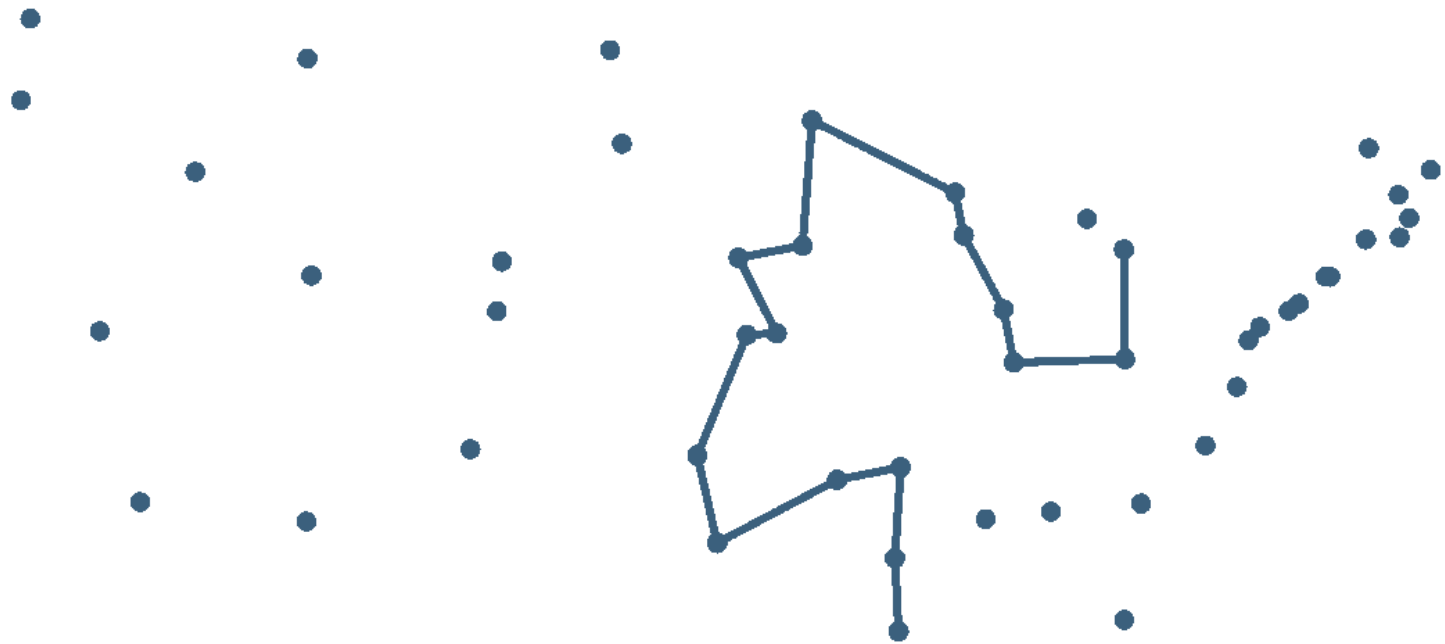
Exemple de l'heuristique Nearest Neighbour



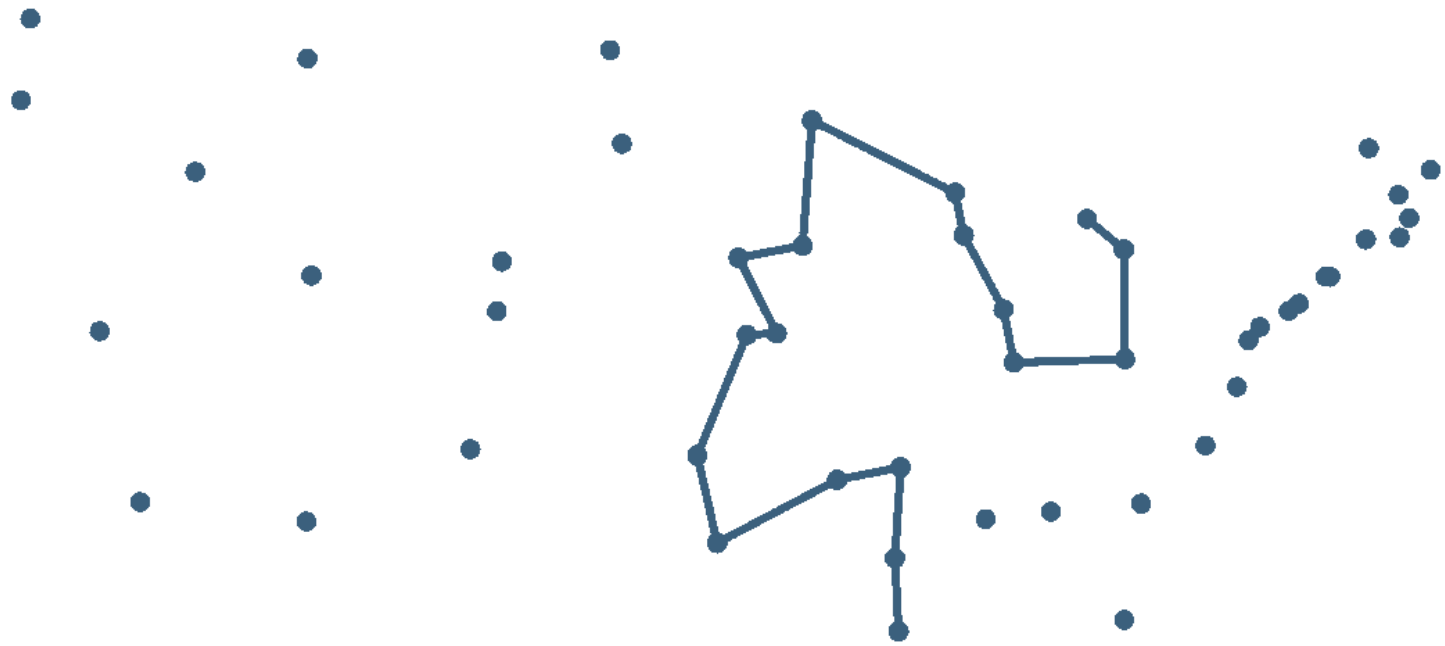
Exemple de l'heuristique Nearest Neighbour



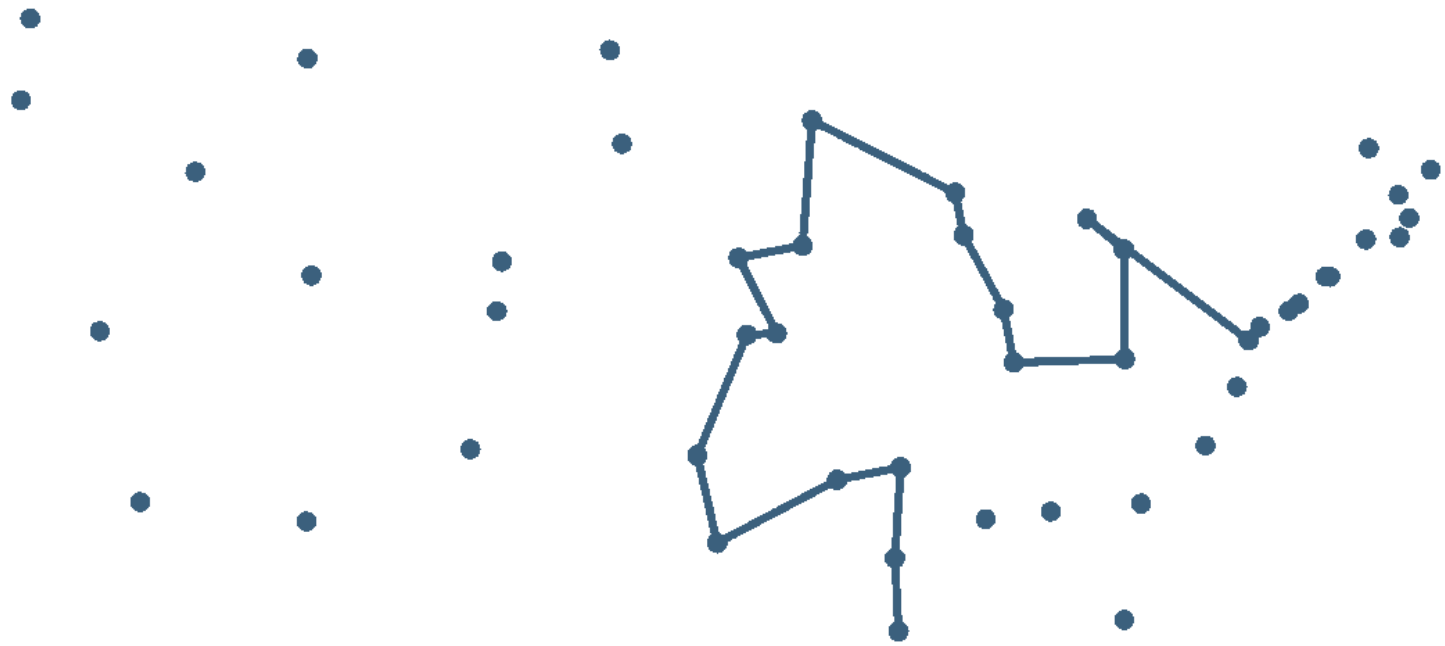
Exemple de l'heuristique Nearest Neighbour



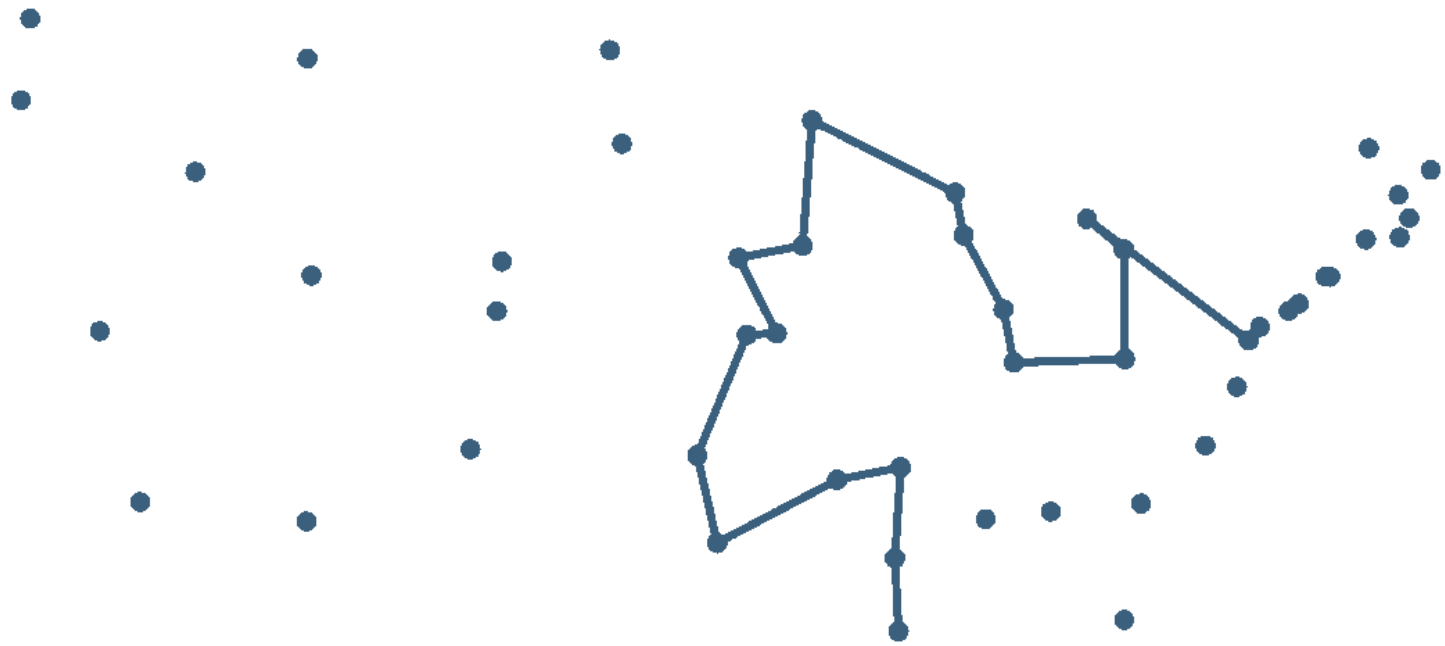
Exemple de l'heuristique Nearest Neighbour



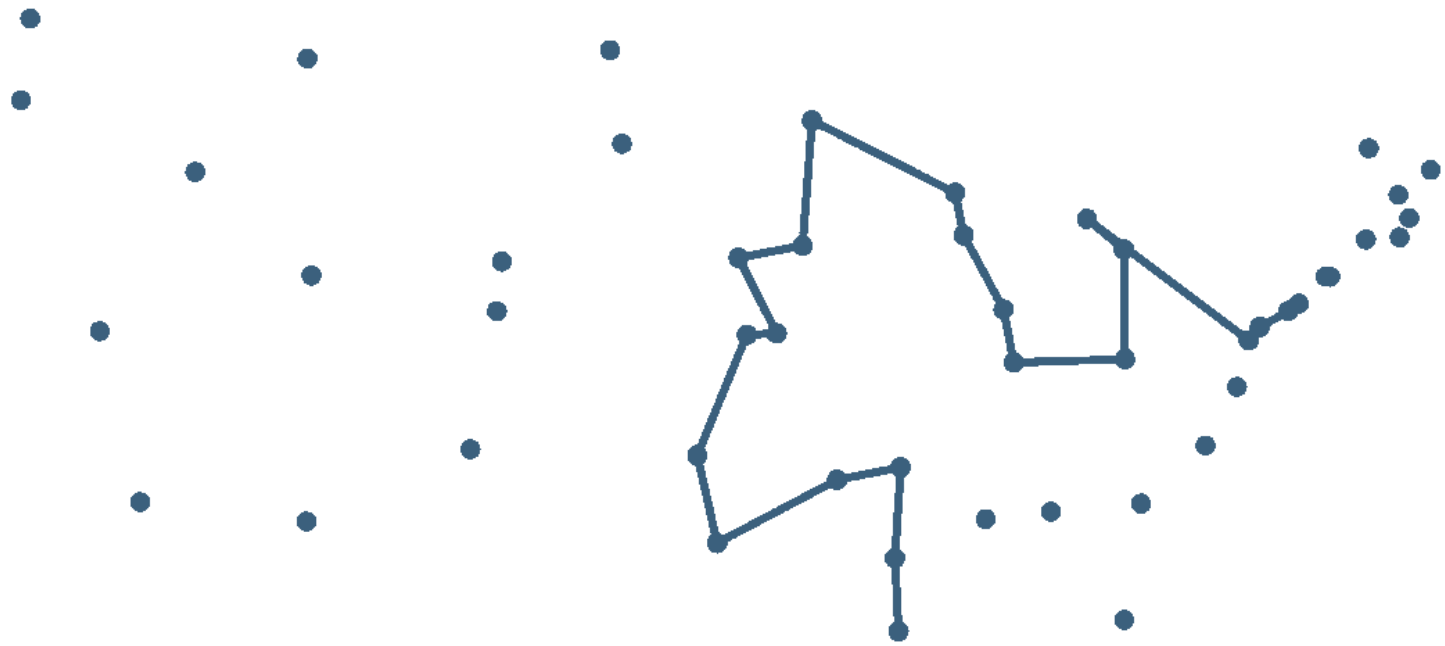
Exemple de l'heuristique Nearest Neighbour



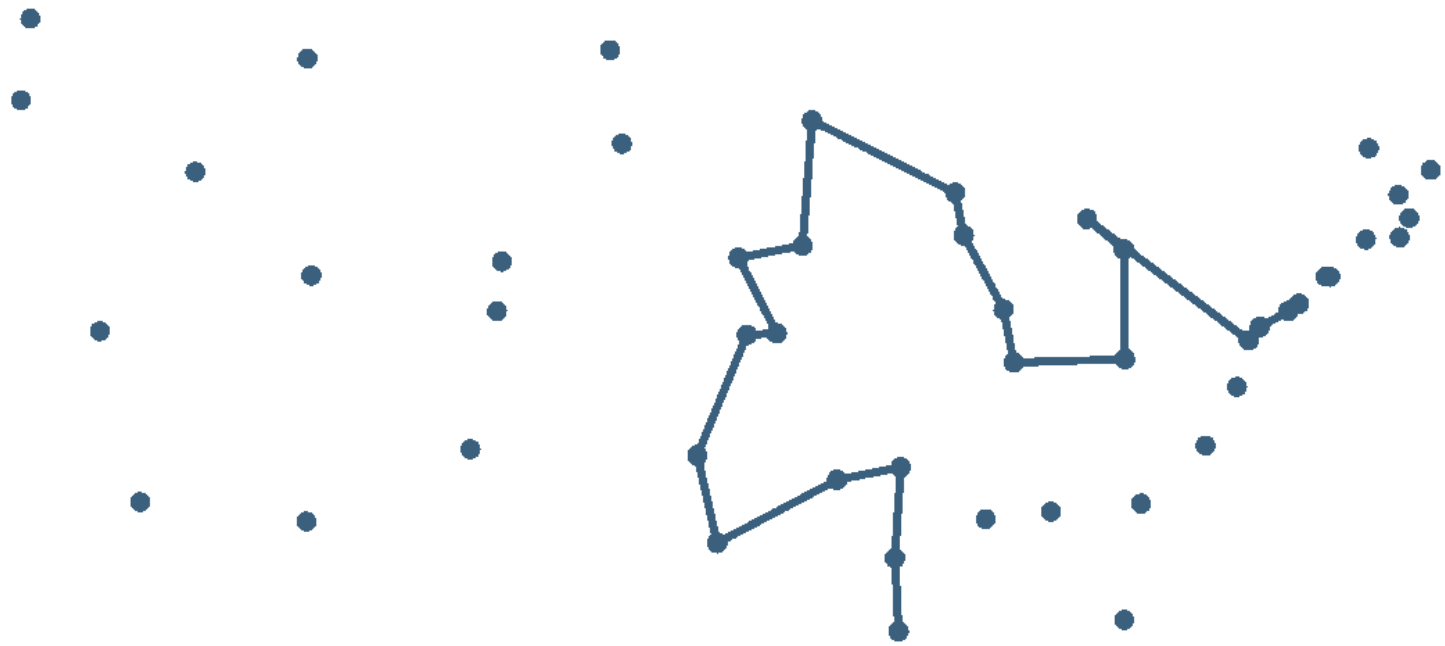
Exemple de l'heuristique Nearest Neighbour



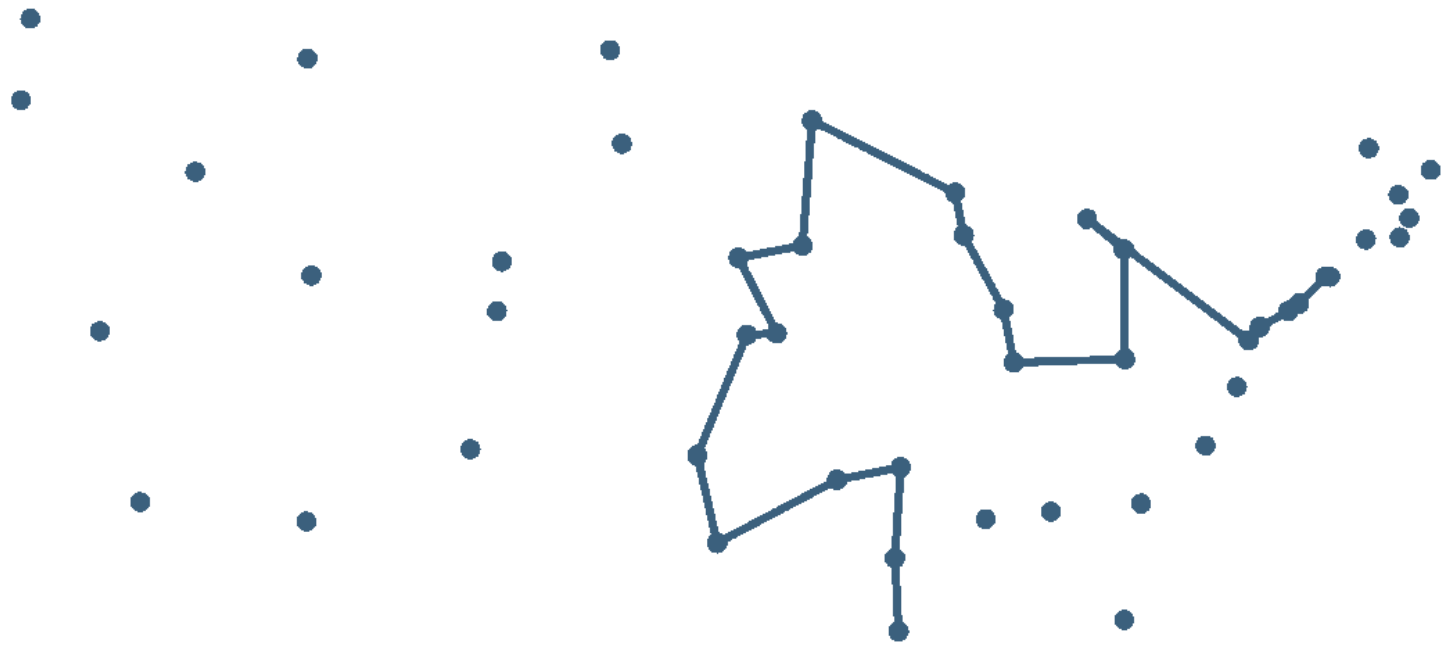
Exemple de l'heuristique Nearest Neighbour



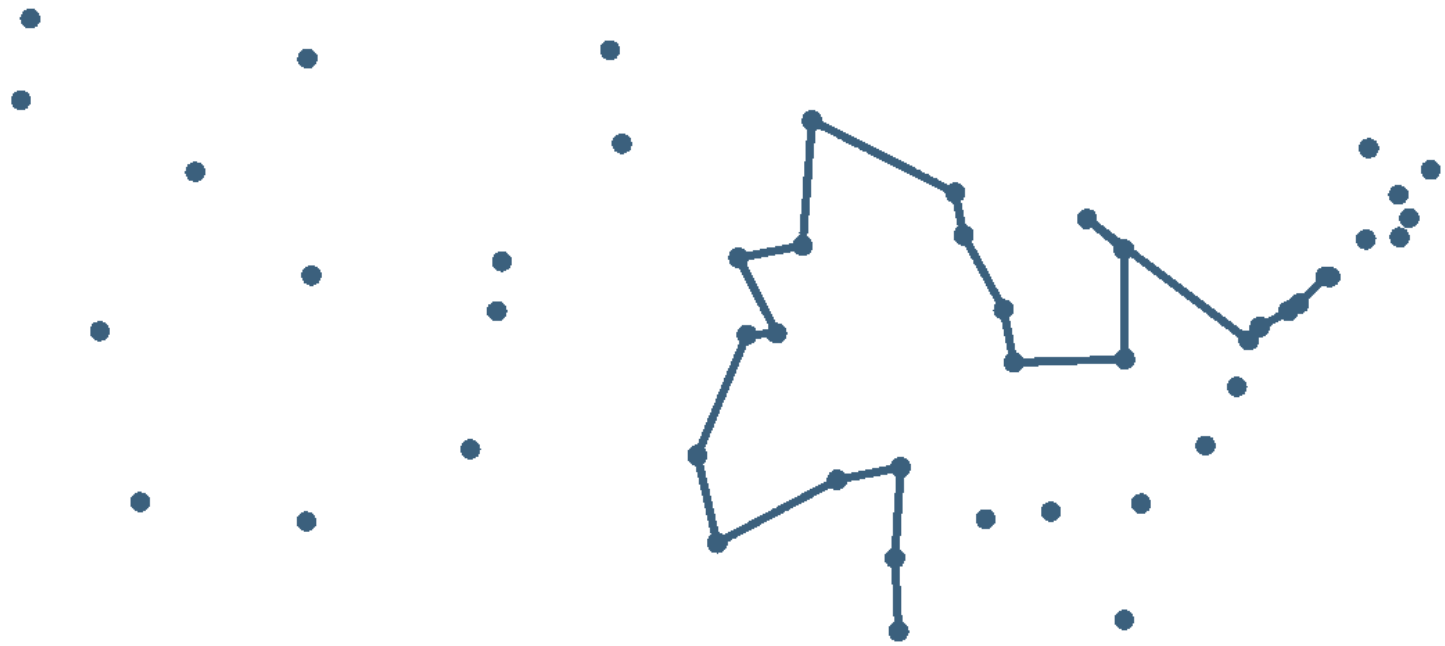
Exemple de l'heuristique Nearest Neighbour



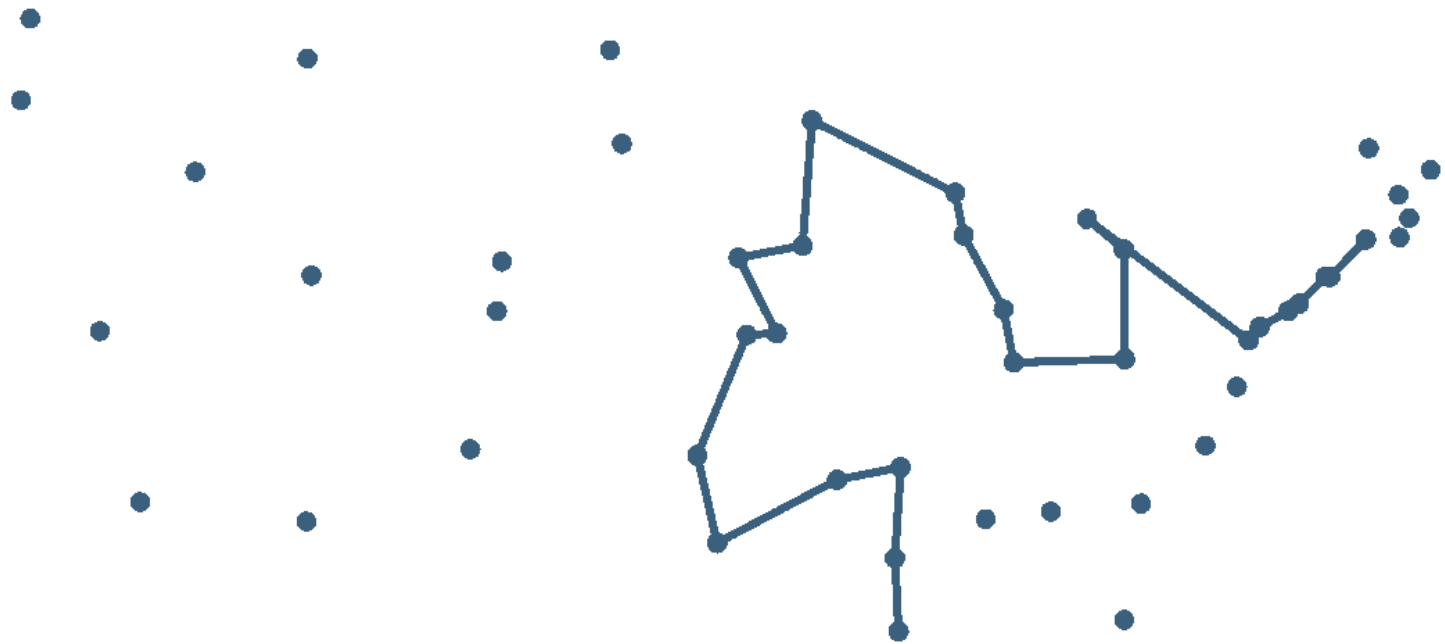
Exemple de l'heuristique Nearest Neighbour



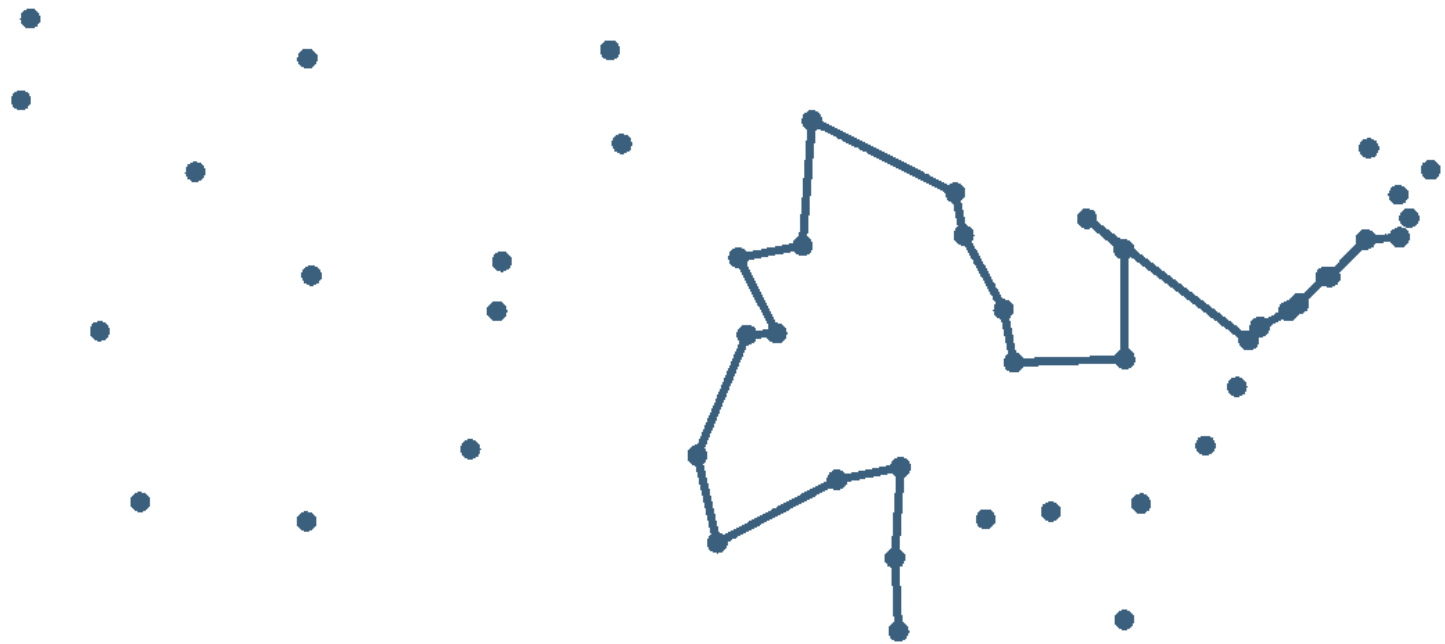
Exemple de l'heuristique Nearest Neighbour



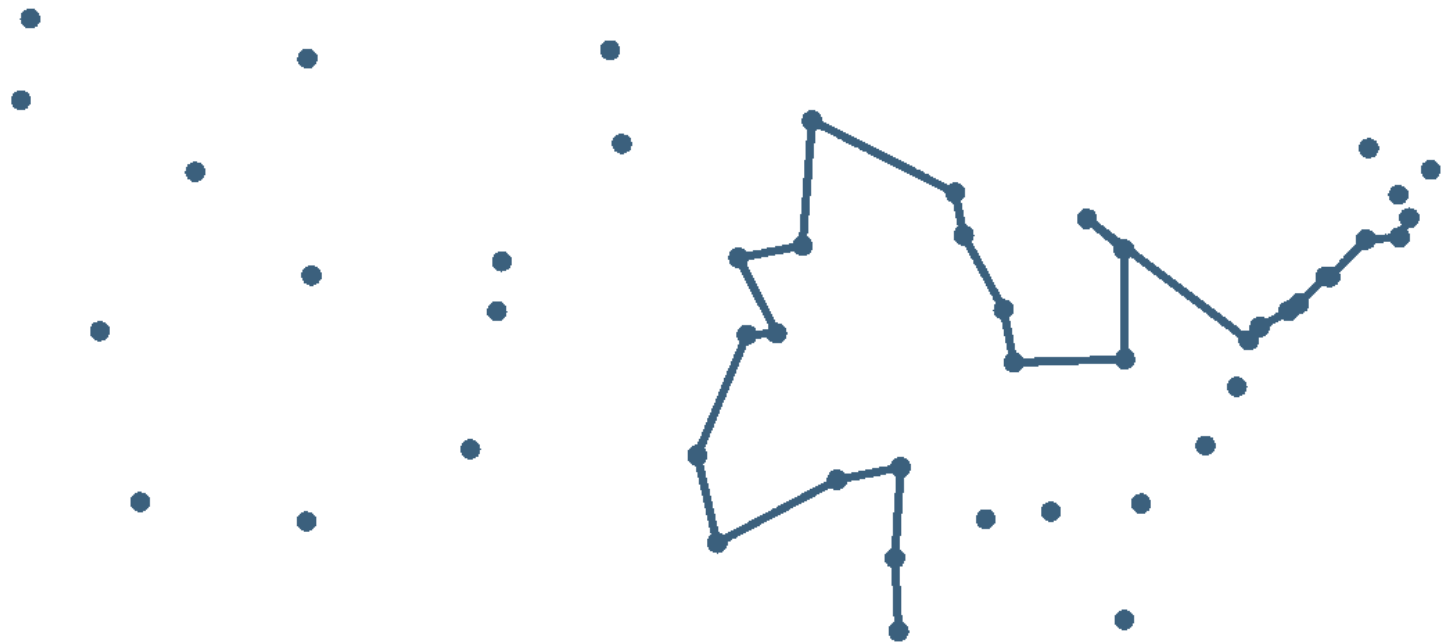
Exemple de l'heuristique Nearest Neighbour



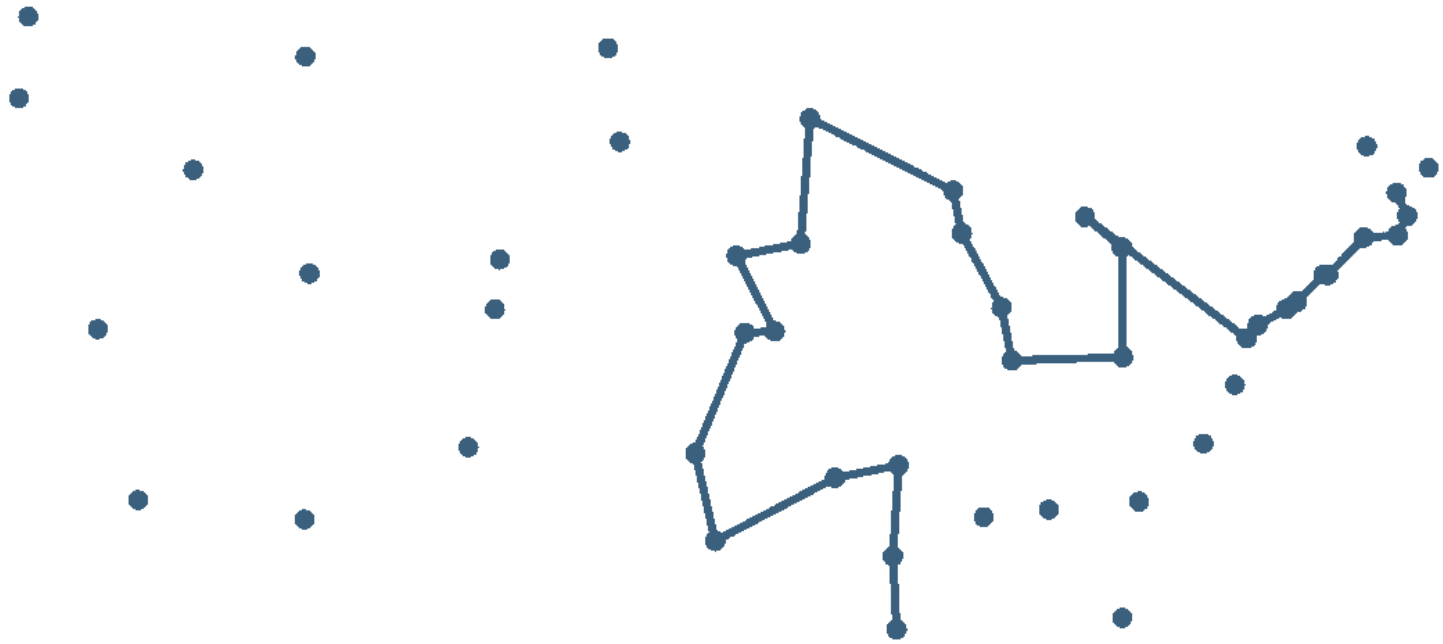
Exemple de l'heuristique Nearest Neighbour



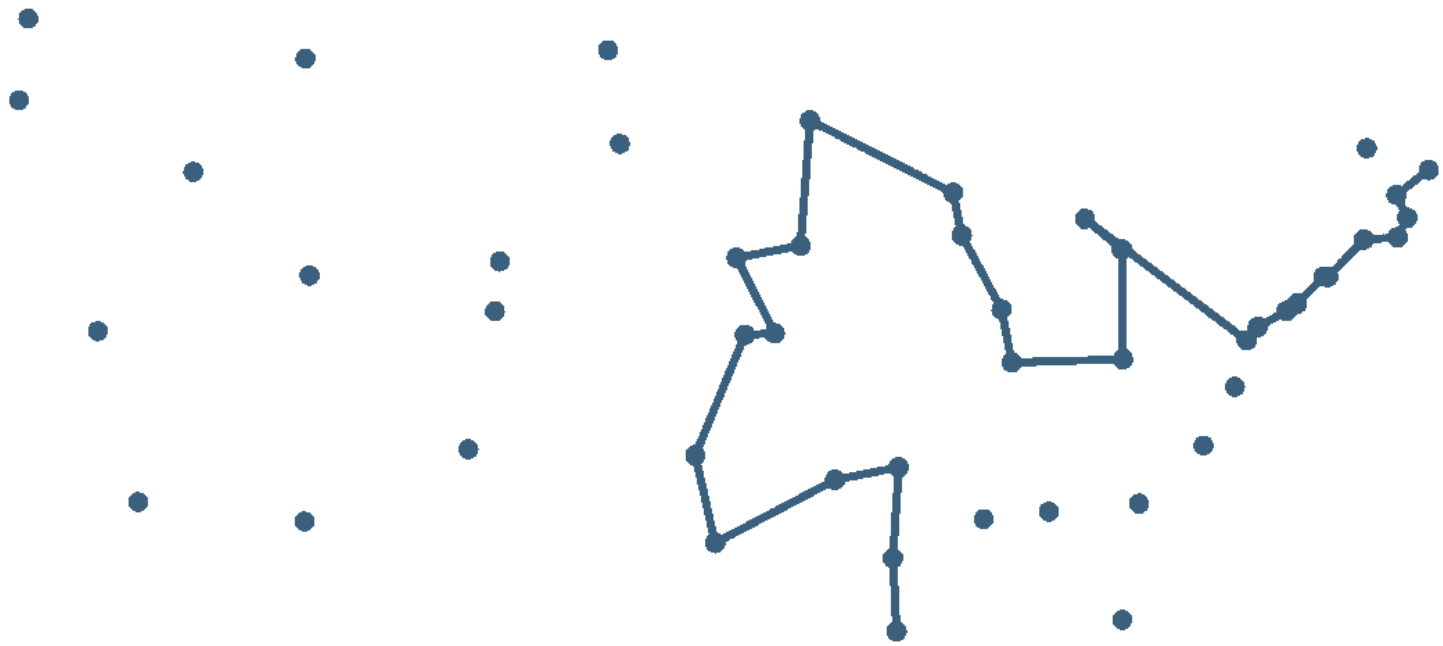
Exemple de l'heuristique Nearest Neighbour



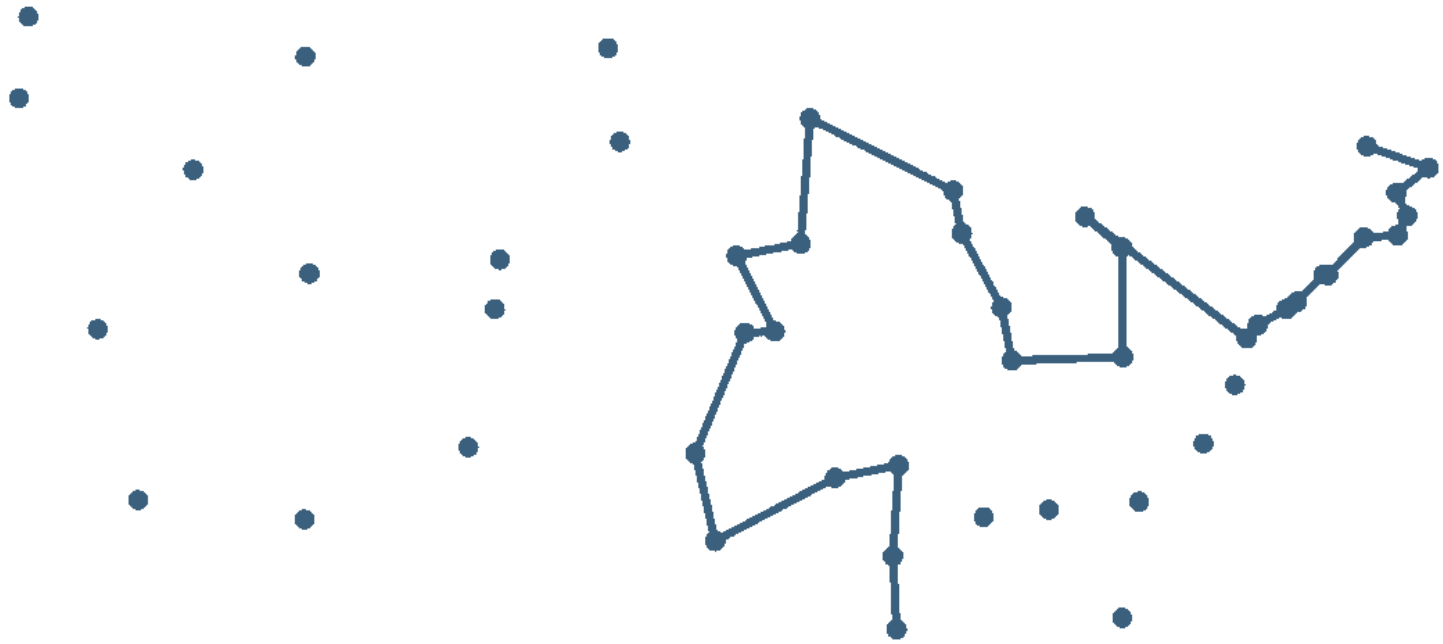
Exemple de l'heuristique Nearest Neighbour



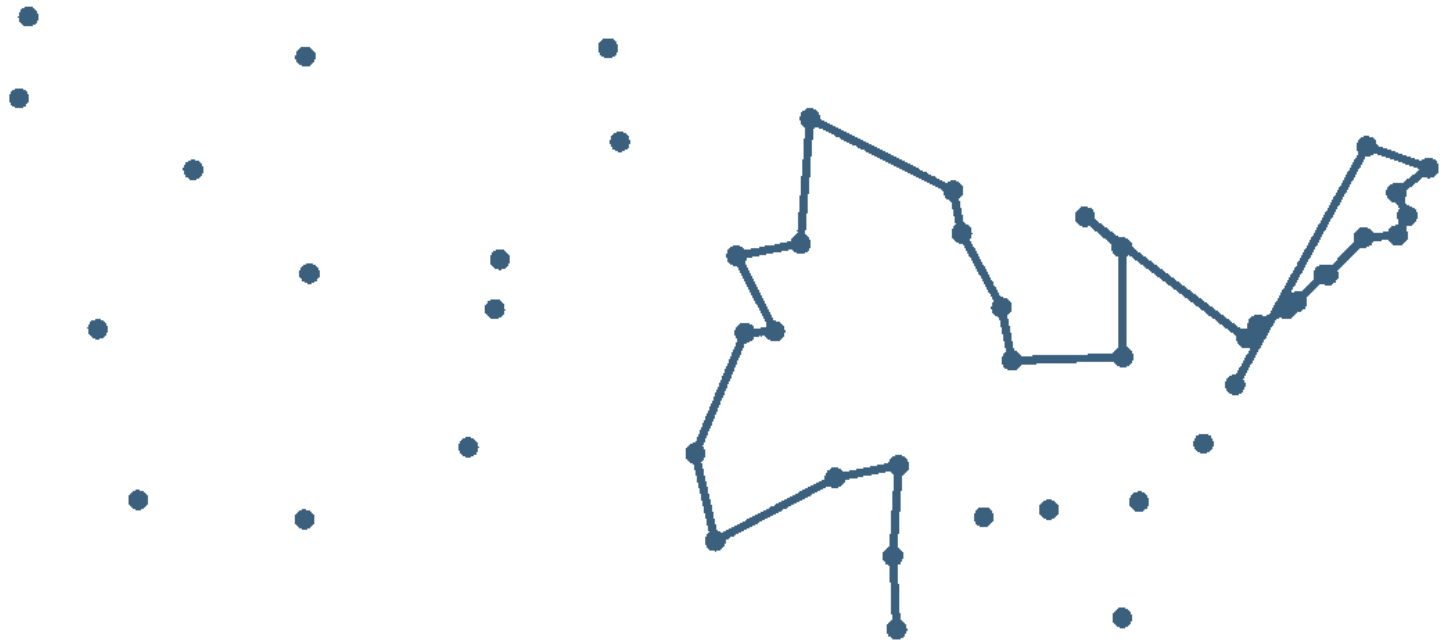
Exemple de l'heuristique Nearest Neighbour



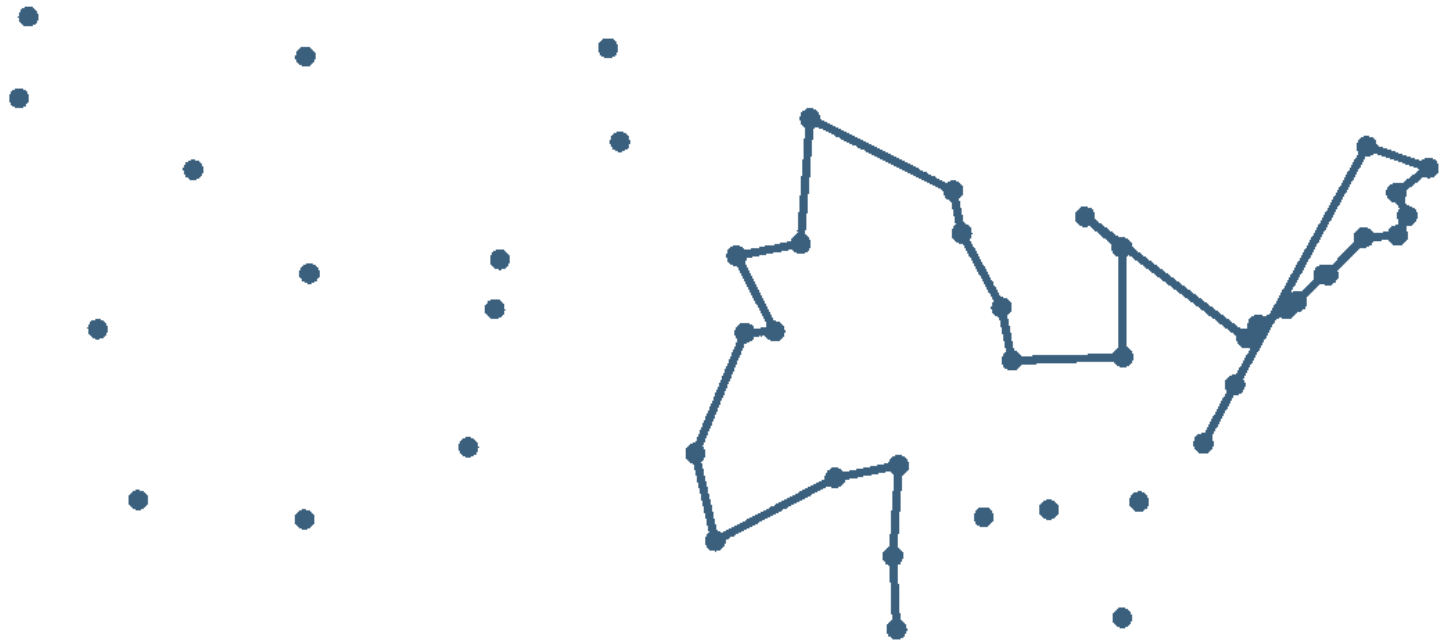
Exemple de l'heuristique Nearest Neighbour



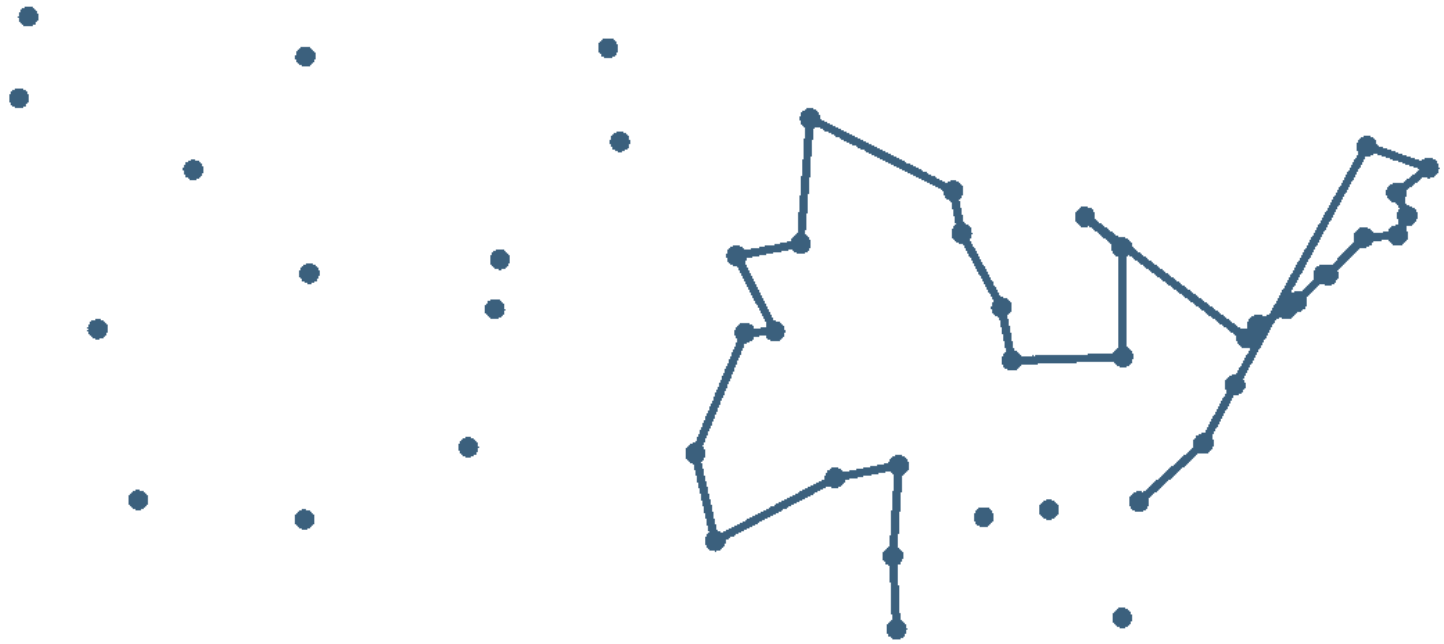
Exemple de l'heuristique Nearest Neighbour



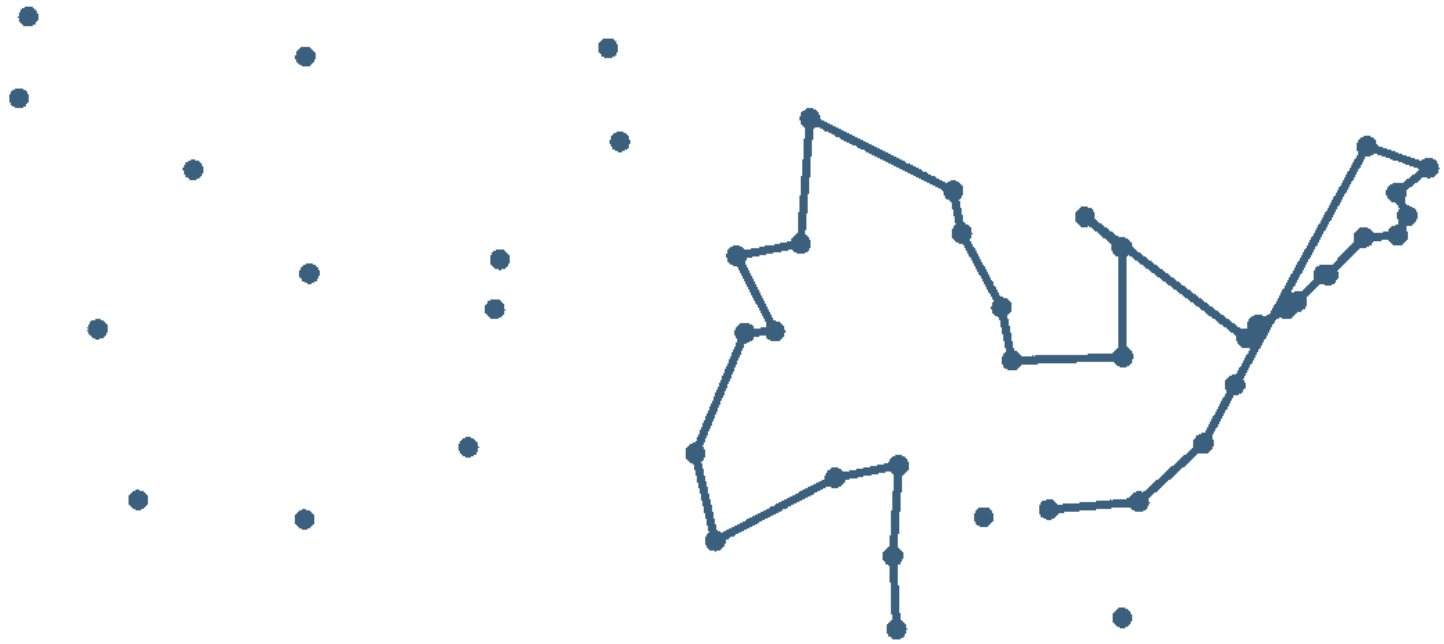
Exemple de l'heuristique Nearest Neighbour



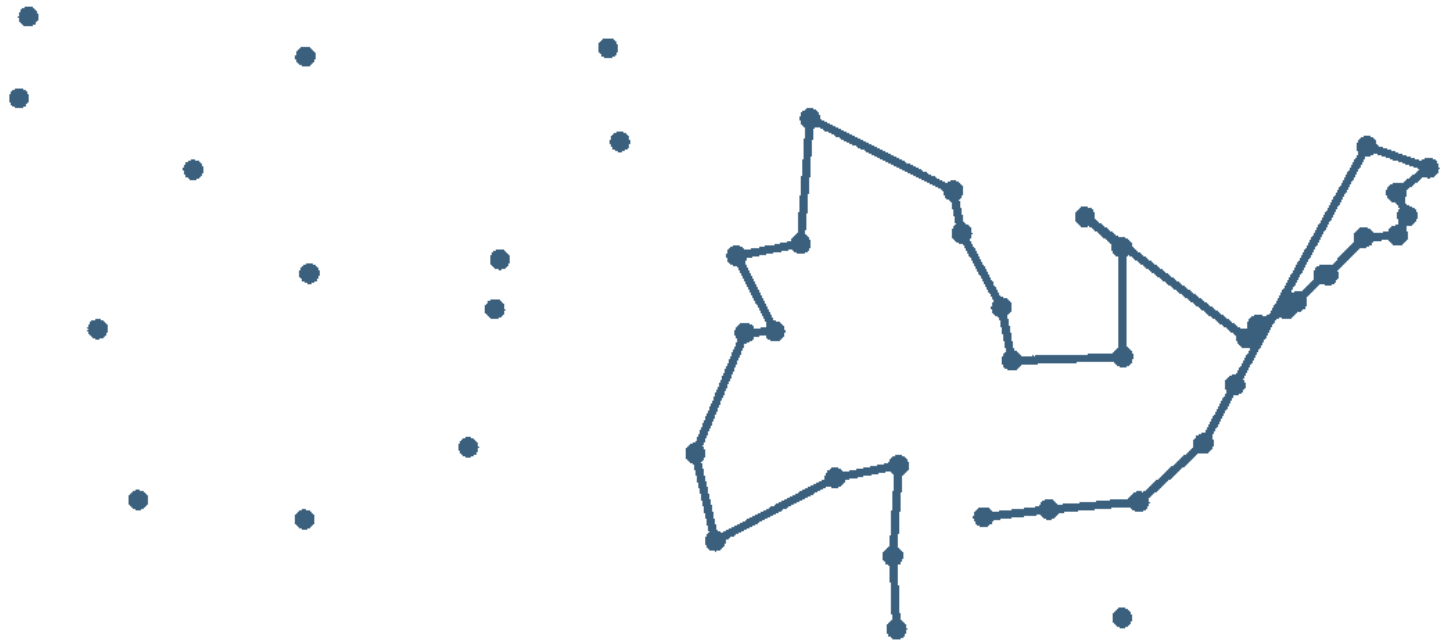
Exemple de l'heuristique Nearest Neighbour



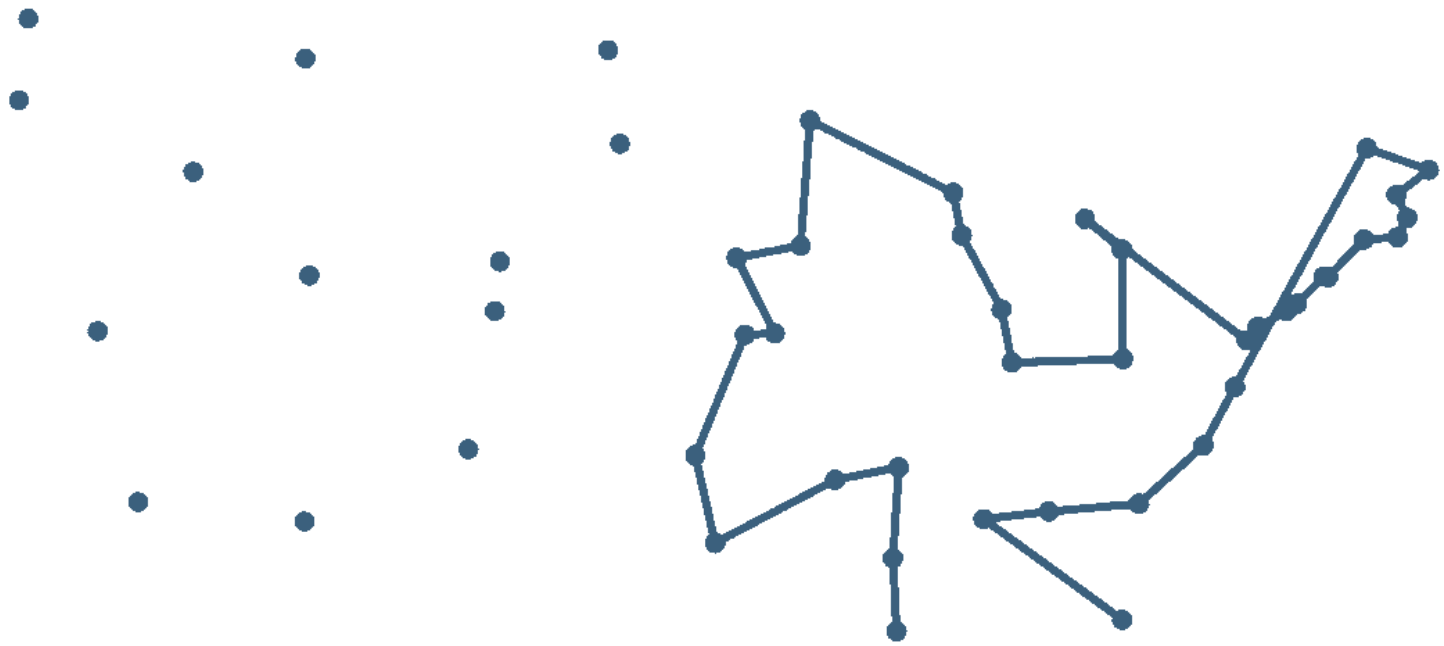
Exemple de l'heuristique Nearest Neighbour



Exemple de l'heuristique Nearest Neighbour



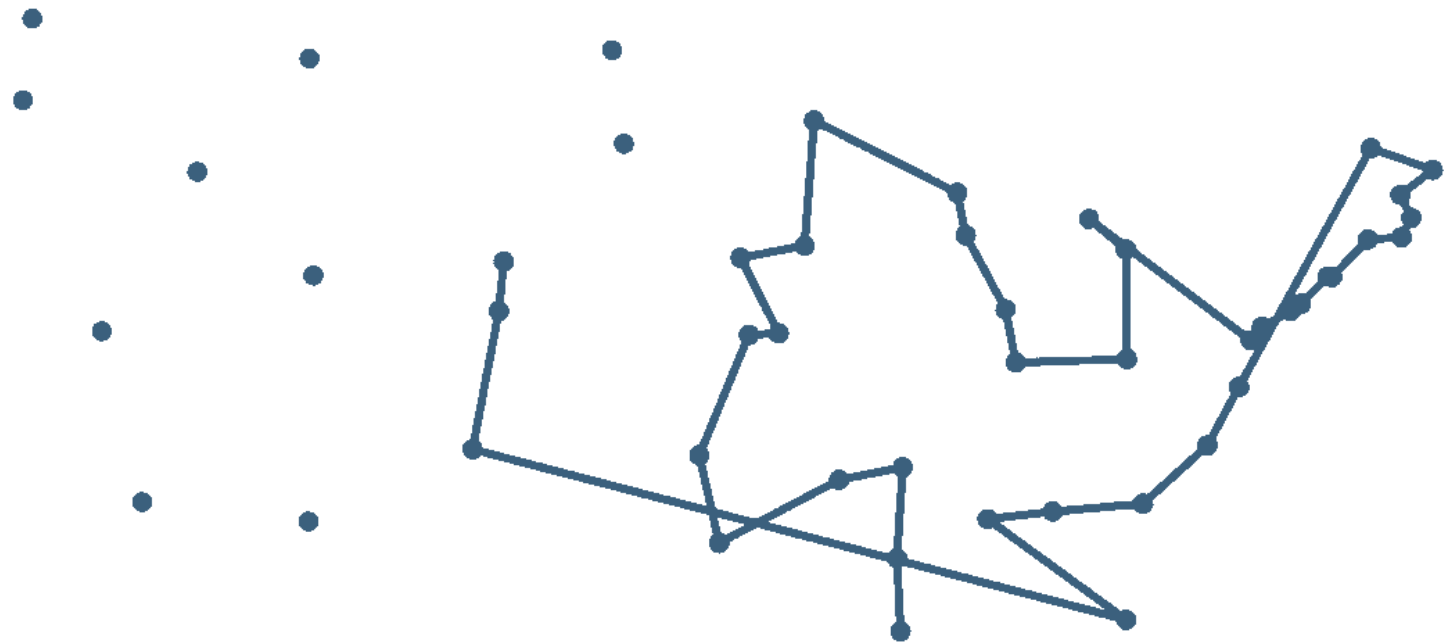
Exemple de l'heuristique Nearest Neighbour



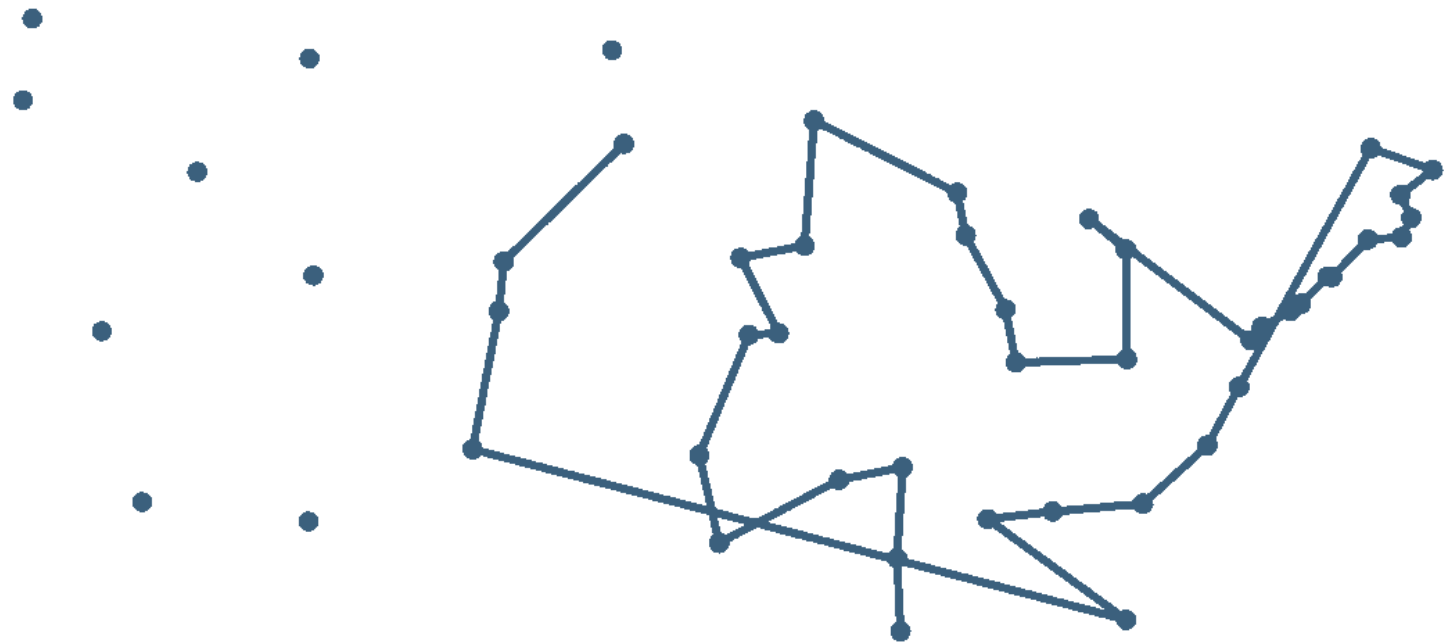
Exemple de l'heuristique Nearest Neighbour

Exemple de l'heuristique Nearest Neighbour

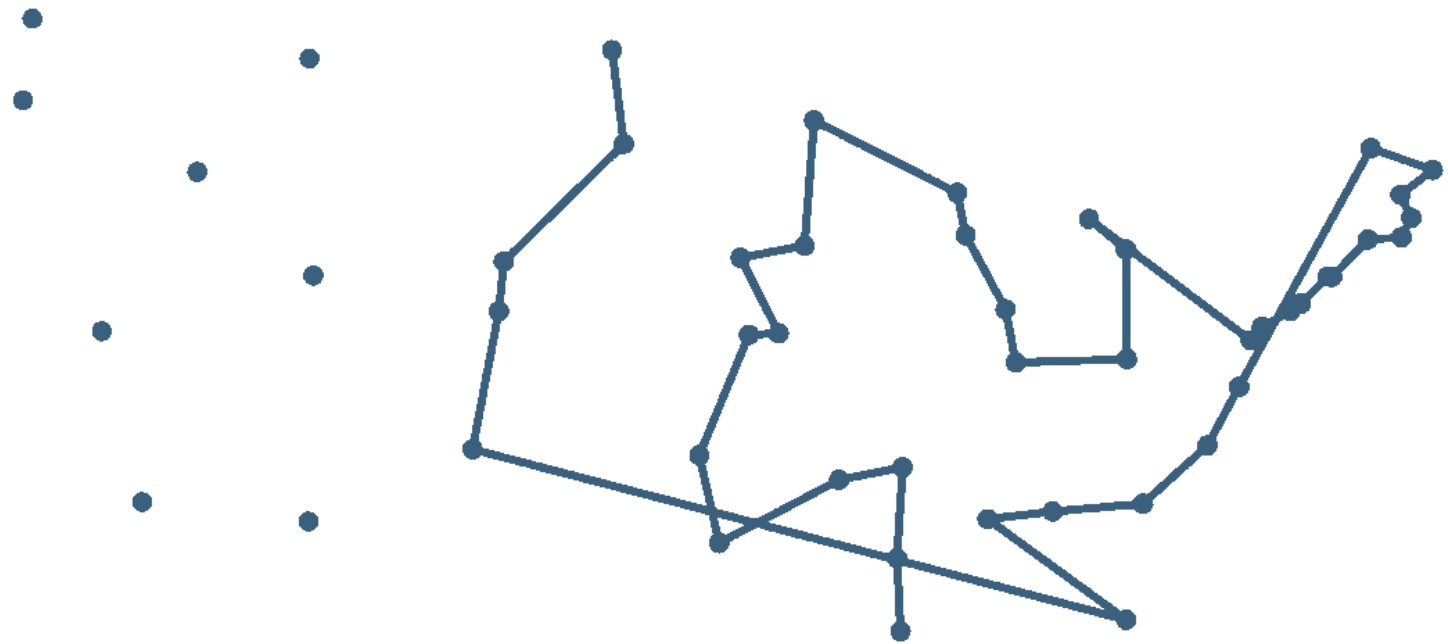
Exemple de l'heuristique Nearest Neighbour



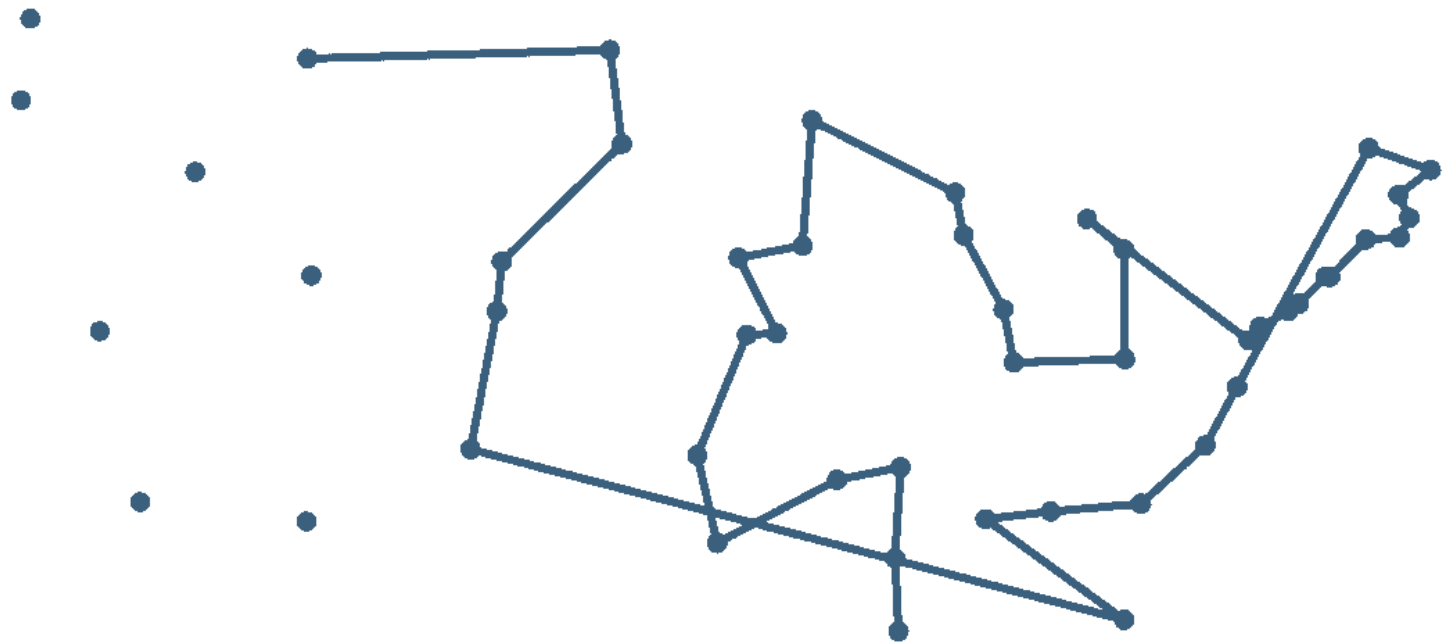
Exemple de l'heuristique Nearest Neighbour



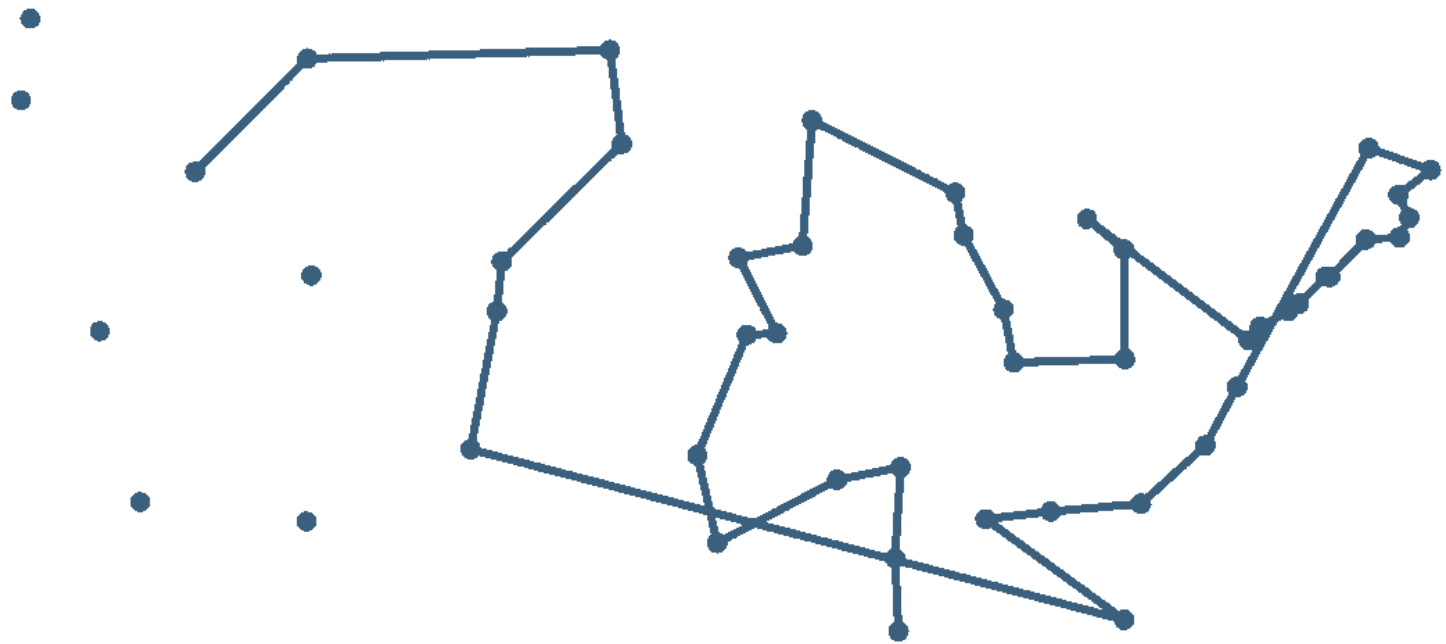
Exemple de l'heuristique Nearest Neighbour



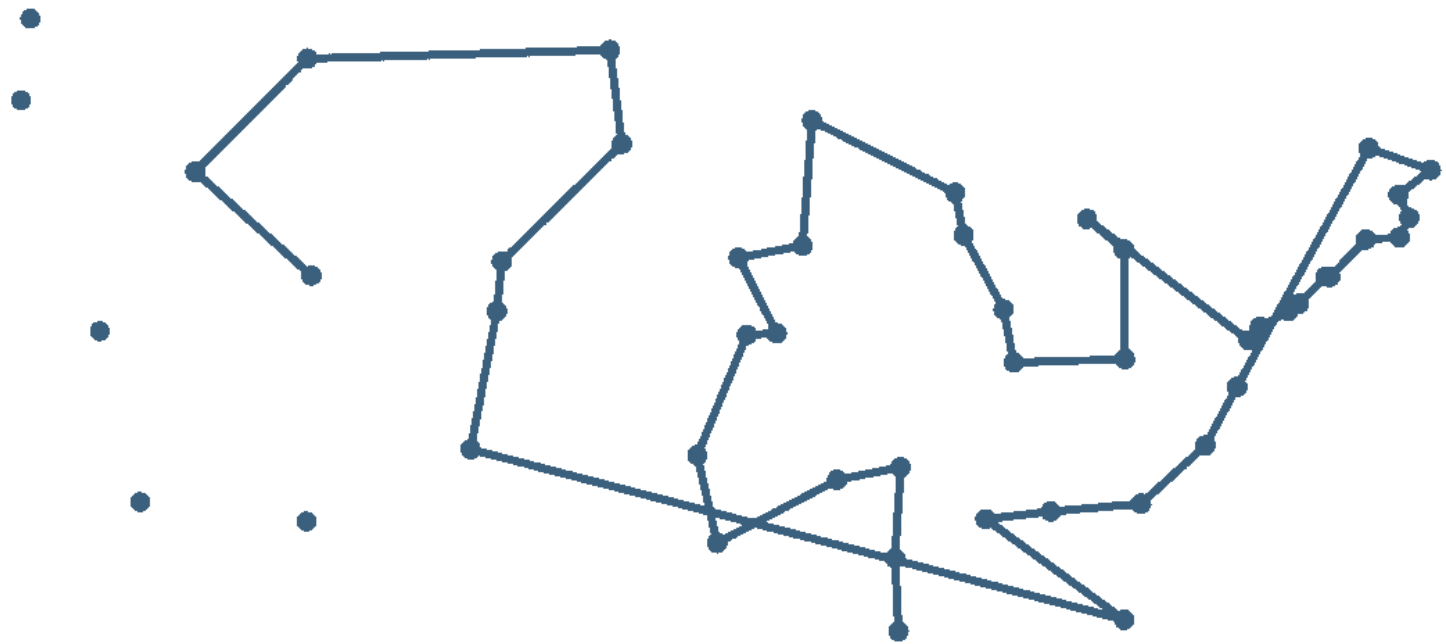
Exemple de l'heuristique Nearest Neighbour



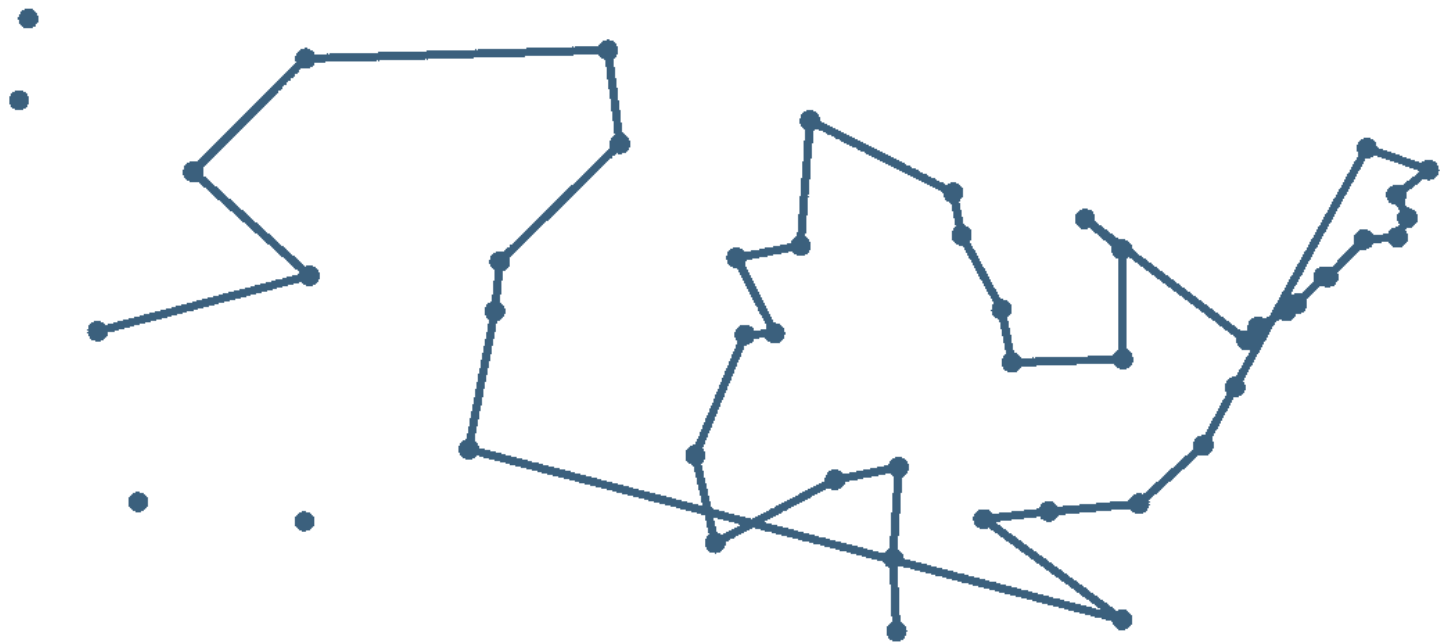
Exemple de l'heuristique Nearest Neighbour



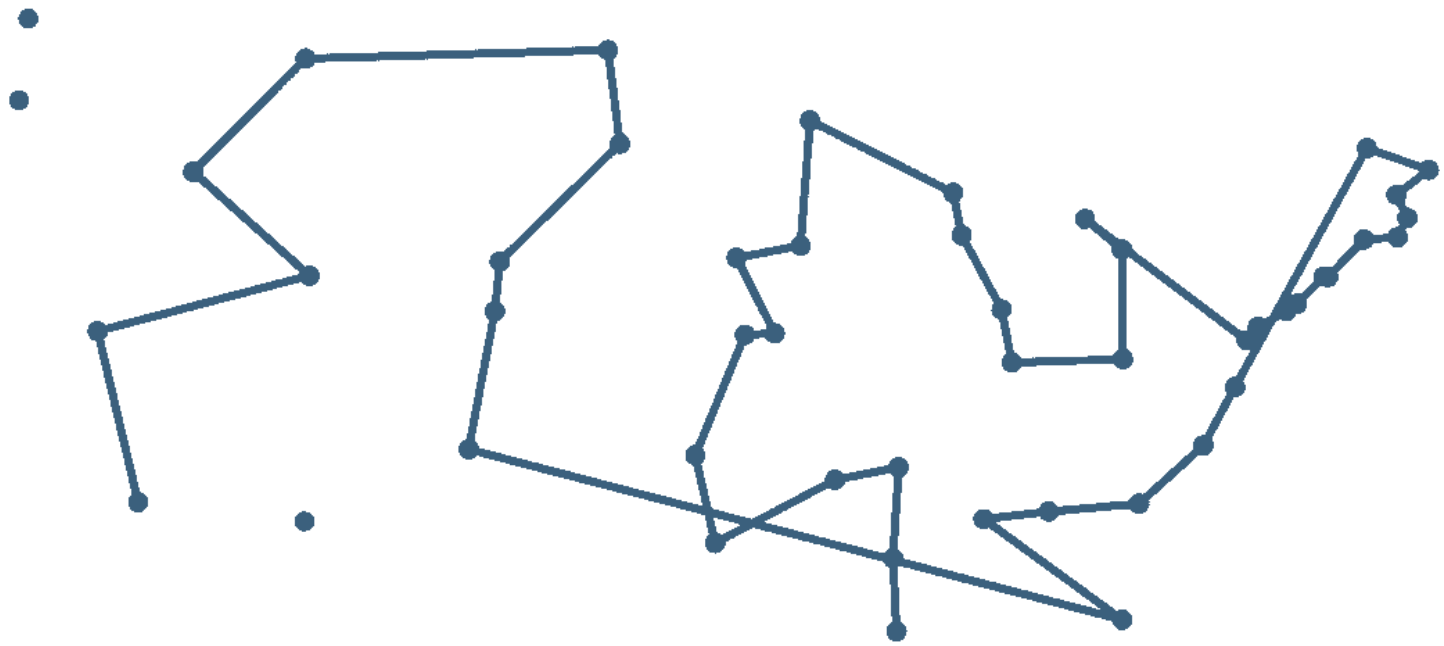
Exemple de l'heuristique Nearest Neighbour



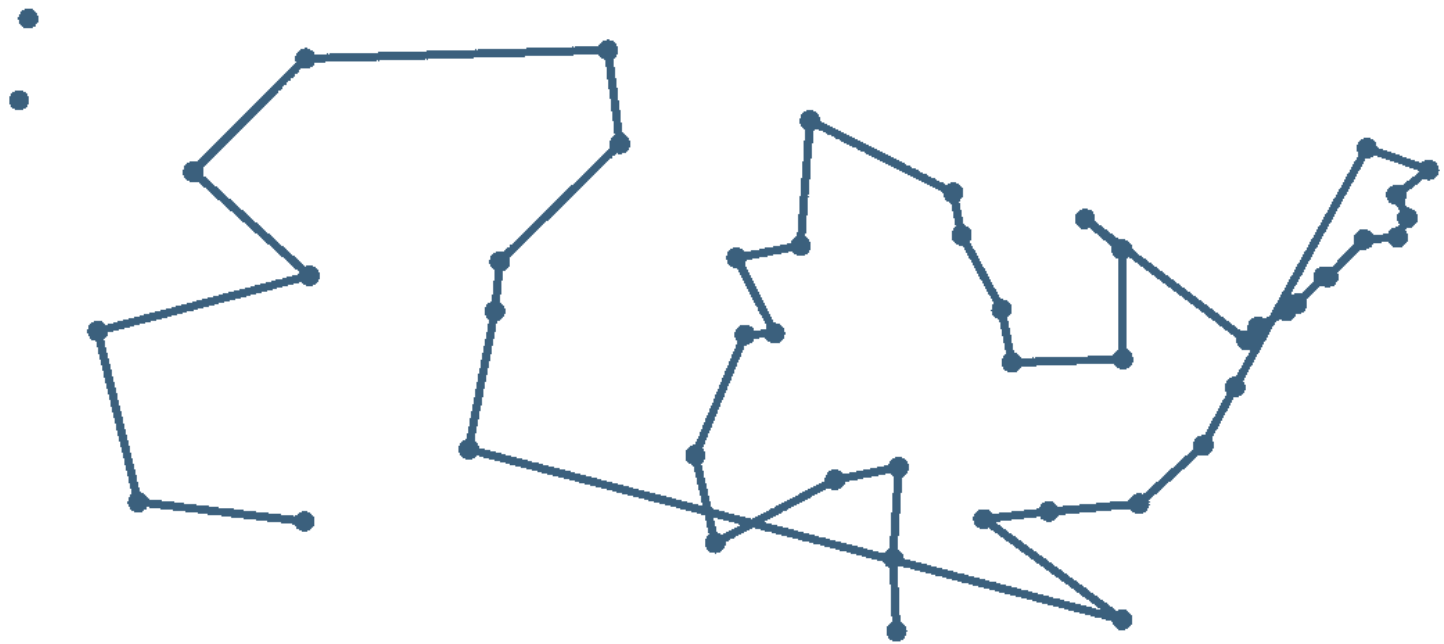
Exemple de l'heuristique Nearest Neighbour



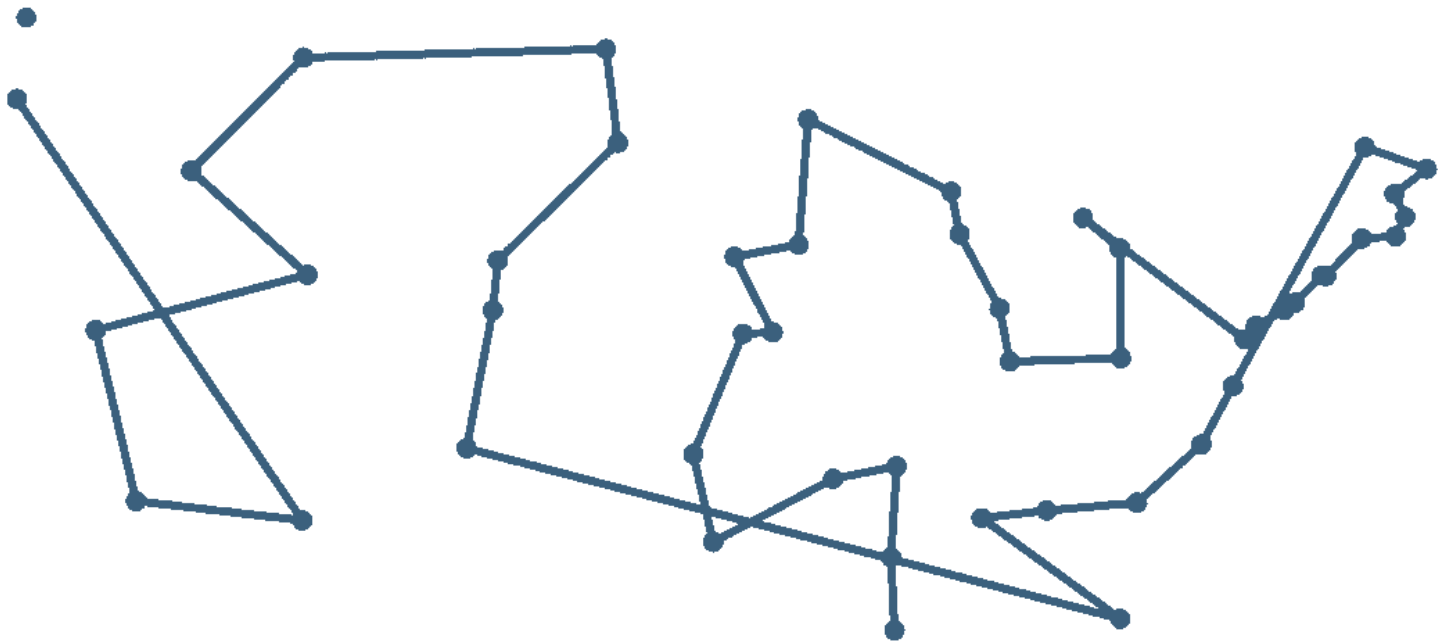
Exemple de l'heuristique Nearest Neighbour



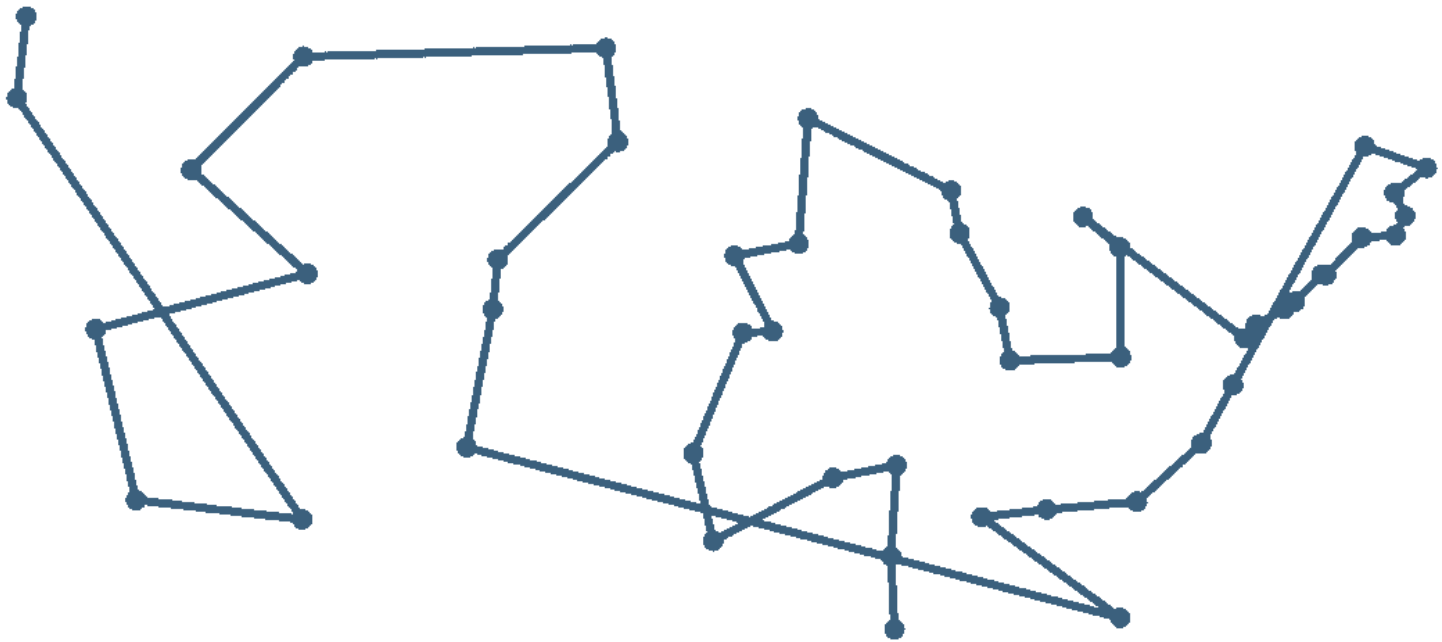
Exemple de l'heuristique Nearest Neighbour



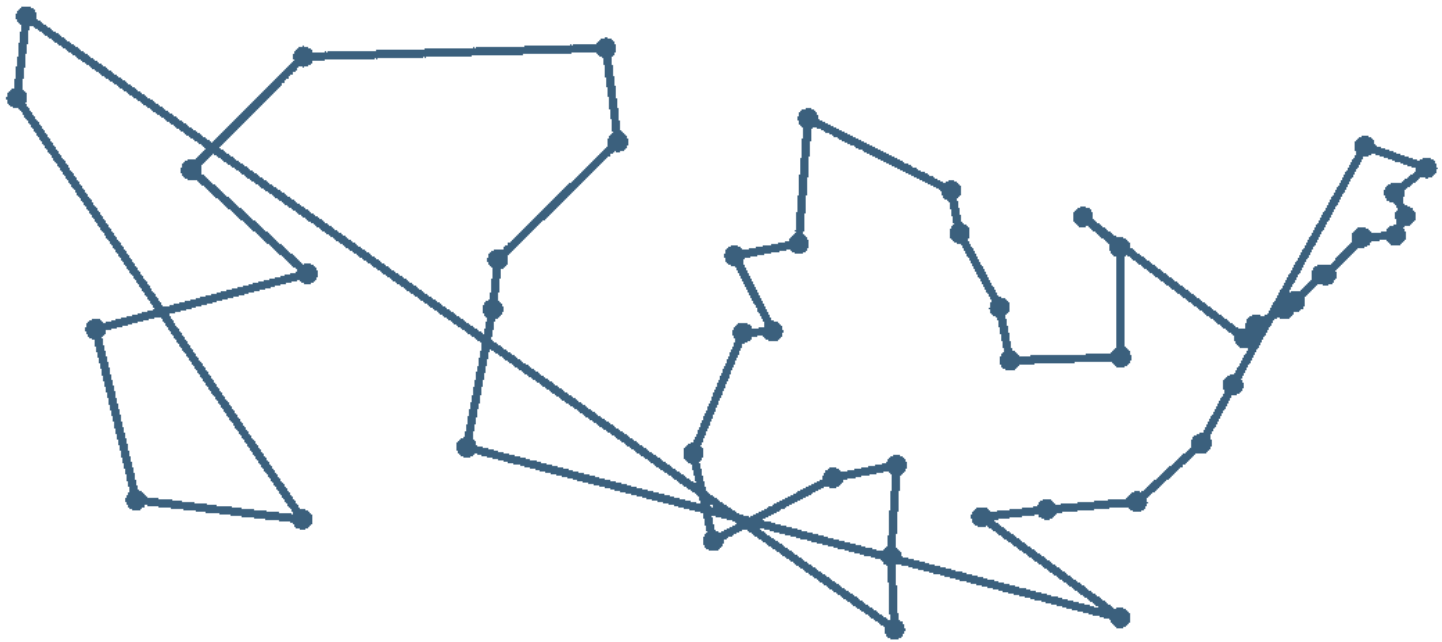
Exemple de l'heuristique Nearest Neighbour



Exemple de l'heuristique Nearest Neighbour

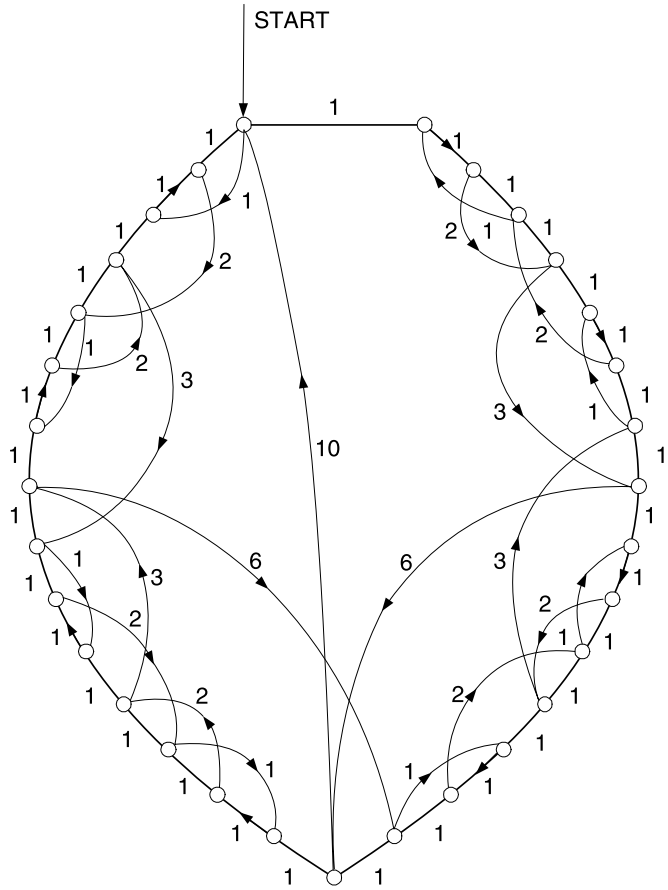


Exemple de l'heuristique Nearest Neighbour



Remarques sur l'heuristique Nearest Neighbour

- Facile à implémenter
- Rapide (complexité $O(n^2)$)
- La solution trouvée peut être (très) loin de l'optimum



Algorithmes d'approximation

Définition

- Face à un problème d'optimisation NP-difficile, on essaie souvent de trouver un *algorithme d'approximation*.
- Étant donné une problème d'optimisation, soit $\mathrm{OPT}(I)$ la solution optimale de l'instance I.
- Soit $\mathcal{A}$ un algorithme (polynomial) qui retourne une solution $\mathcal{A}(I)$ de l'instance I.
- Alors, le *facteur d'approximation* de $\mathcal{A}$ est défini comme

$$\rho = \max_I \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)}.$$

- On dit que $\mathcal{A}$ est un algorithme de ρ-*approximation*.

TSP n'est pas approximable (1/3)

Théorème (Sahni and Gonzalez 1976)

À moins que P=NP, il n'y a pas d'algorithme de ρ-approximation pour TSP pour tout $\rho \geq 1$.

Démonstration (suite)

- Supposons par l'absurde qu'il existe un algorithme de ρ-approximation A pour TSP.
- Nous prouverons qu'il existe un algorithme polynomial pour le problème du cycle hamiltonien.
- Comme ce dernier est NP-complet, cela implique P = NP.

TSP n'est pas approximable (2/3)

Démonstration (suite)

- Étant donné une graphe G, nous construisons une instance de TSP avec $n = |V(G)|$ villes : les distances sont données par

$$c_{ij} = \begin{cases} 1 & \text{if } ij \in E(G) \\ 2 + (\rho - 1)n & \text{sinon} \end{cases}$$

$n - 1 + 2 + (\rho - 1)n$

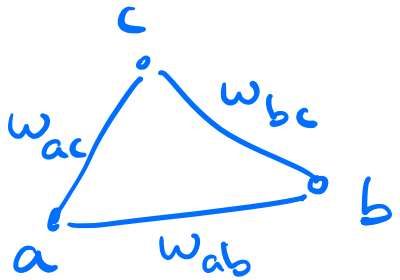
- Maintenant nous appliquons A à cette instance.
- Si le tour trouvé est de longueur n, alors ce tour est un cycle hamiltonien dans G.
- Sinon le tour trouvé est de longueur au moins $n + 1 + (\rho - 1)n = \rho n + 1$.

TSP n'est pas approximable (3/3)

Démonstration (suite)

- Soit $OPT(K_n, c)$ la longueur d'un tour optimal, alors $\frac{\rho n+1}{OPT(K_n,c)} \leq \rho$ puisque A est un algorithme de ρ-approximation.
- Donc, $OPT(K_n, c) > n$, ce qui montre que G n'a pas de cycle hamiltonien. □

Algorithme d'approximation pour le voyageur de commerce métrique



- Il existe des algorithmes d'approximation lorsque les poids sur les arêtes satisfont l'inégalité triangulaire : soient a, b et c des sommets de G quelconques, alors $w_{ac} \leq w_{ab} + w_{bc}$.
- On parle dans ce cas du problème de voyageur de commerce (TSP) *métrique*.

Un algorithme d'approximation pour le TSP métrique

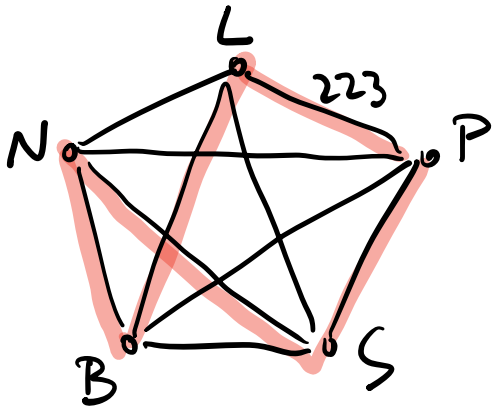
(Kruskal ou Prim)

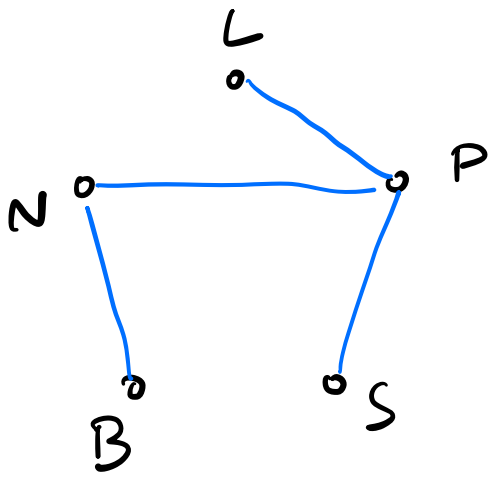
1. Trouver un arbre couvrant T de poids minimum dans $G = (V, E)$.
2. Construire le multigraphe H avec ensemble de sommets V et deux copies de chaque arête de T.
3. Trouver un cycle eulérien $C = (v_1, e_1, v_2, e_2, \ldots, v_{n-1}, e_{n-1}, v_n, e_n, v_1)$ dans H.
4. Prendre les sommets de C dans l'ordre et supprimer tout sauf la première occurrence de chaque sommet (en gardant aussi la dernière occurence de v_1).

Illustration de l'algorithme

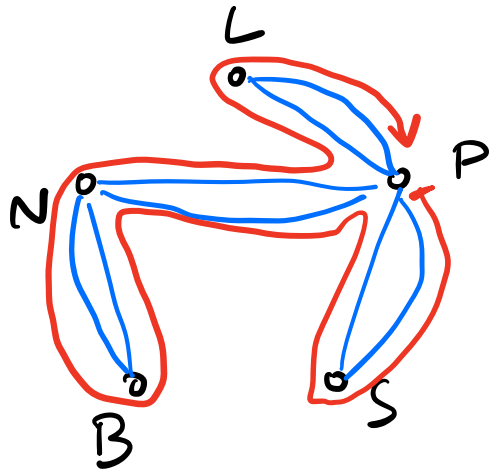
	Bordeaux	Lille	Nantes	Paris	Strasbourg
Bordeaux	×	792	329	561	958
Lille	792	×	593	223	524
Nantes	329	593	×	393	832
Paris	561	223	393	×	448
Strasbourg	958	524	832	448	×

P, S, ~~P~~, N, B, ~~N~~, ~~P~~, L, P





T



H

Complexité de l'algorithme

Remarque

La complexité de l'algorithme Double-Tree est de $O(n^2 \log n)$.

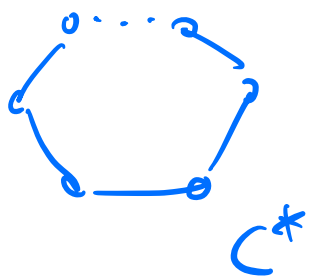
1. trouver un arbre couvrant T de poids minimum : $O(n^2 \log n)$ (Kruskal)
2. construire le multigraphe H : $O(n)$
3. trouver un cycle eulérien C : $O(n)$ (Hierholzer)
4. supprimer des sommets de C : $O(n)$

Le facteur d'approximation de Double-Tree (1/2)

Theorem

Double-Tree est un algorithme de 2-approximation pour le TSP métrique.

- Soit C^* un plus court tour de voyageur de commerce, de longueur OPT.
- Si l'on supprime une arête quelconque de C^*, on obtient un graphe couvrant connexe et acyclique, c'est-à-dire, un arbre couvrant de poids au plus OPT.
- En particulier, si T est un arbre couvrant de poids minimum, alors $w(T) \leq w(C^*) = \text{OPT}$.
- Donc, $w(H) = 2w(T) \leq 2\text{OPT}$.



Le facteur d'approximation de Double-Tree (2/2)

- Supposons que le sommet v_i est supprimé lors de la dernière étape de l'algorithme Double-Tree.
- Soient v_{i-1} son prédécesseur et v_{i+1} son successeur dans le cycle.
- Par l'inégalité triangulaire, $w(v_{i-1}v_{i+1}) \leq w(v_{i-1}v_i) + w(v_iv_{i+1})$.
- Donc, aucune suppression d'un sommet dans la dernière étape ne peut augmenter le poids du cycle.
- En particulier, si C' est le cycle final, alors $w(C') \leq w(C) = w(H) \leq 2\text{OPT}$.

Une amélioration de Double-Tree ?

L'algorithme utilise les idées suivantes :

- construire le multigraphe H eulérien sur $V(G)$ en utilisant les arêtes de G (en se permettant de "dédoubler' des arêtes).
- faire des "raccourcis" pour obtenir un tour de voyageur de commerce

Il y a une façon plus fine de trouver le graphe H.

L'algorithme de Christofides

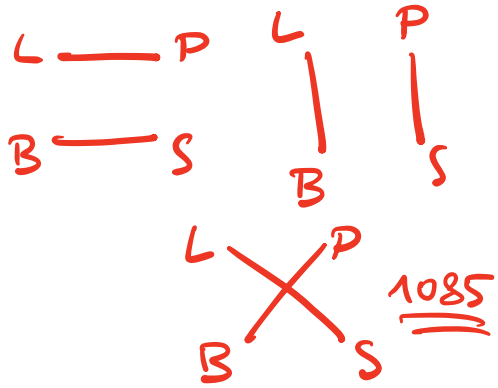
1. Trouver un arbre couvrant T de poids minimum dans G
2. Trouver l'ensemble $U \subseteq V$ de sommets de degré impair dans T
3. Trouver un couplage parfait M de poids minimum dans $G[U]$
4. Construire un graphe eulérien H en ajoutant les arêtes de M à T
5. Trouver un cycle eulérien C' de H
6. Faire des "raccourcis" pour obtenir un cycle hamiltonien $C \subseteq G$ (comme dans l'algorithme *Double-Tree*).

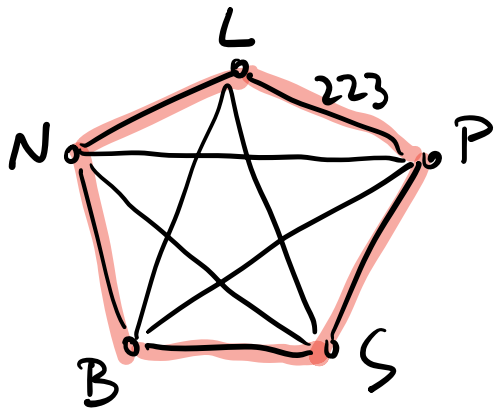
Remarques sur l'algorithme de Christofides

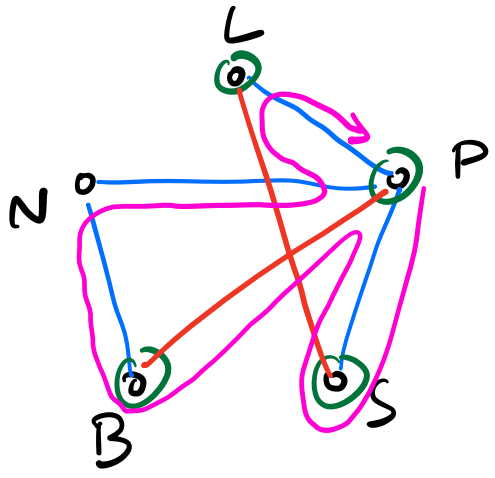
- Pourquoi existe-t-il toujours un couplage parfait dans $G[U]$?
- Il existe un algorithme pour trouver un couplage parfait de poids minimum de complexité $O(n^3)$ (pas vu dans ce cours).
- Pourquoi est-ce un algorithme de 1.5-approximation ?

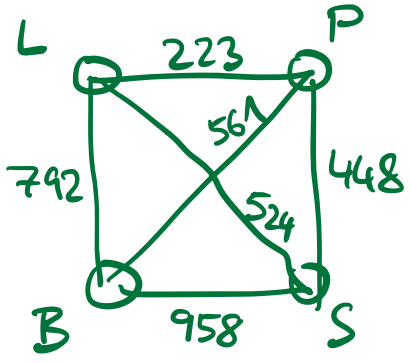
Illustration de l'algorithme

	Bordeaux	Lille	Nantes	Paris	Strasbourg
Bordeaux	×	792	329	561	958
Lille	792	×	593	223	524
Nantes	329	593	×	393	832
Paris	561	223	393	×	448
Strasbourg	958	524	832	448	×









P, S, P, B, N, P, L, P

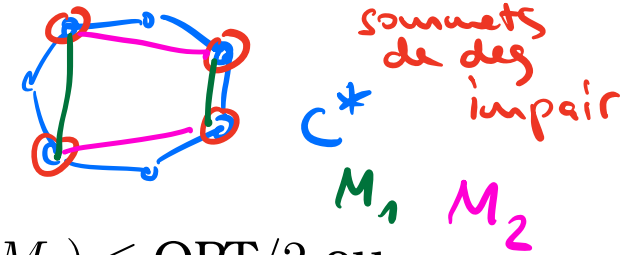
T

Le facteur d'approximation de Christofides

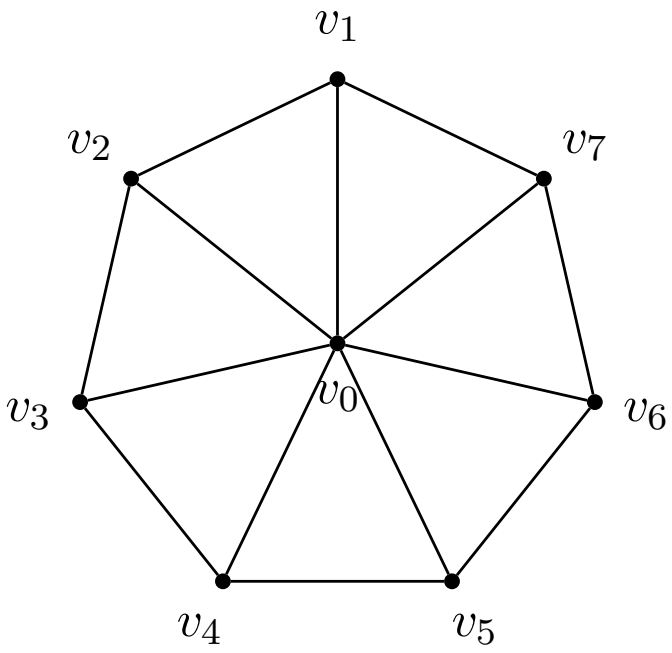
Theorem

Christofides est un algorithme de 1.5-approximation pour le TSP métrique.

- Comme dans l'algorithme Double-Tree, on sait que $w(T) \leq \text{OPT}$.
- Il faut démontrer que $w(M) \leq \text{OPT}/2$.
- Soit C^* un cycle hamiltonien dans G de poids OPT.
- Soient $u_1, u_2, \ldots, u_{2k}$ les sommets de degré impair dans T, numérotés dans le sens de C^*.
- Soit $M_1 = \{u_1u_2, u_3u_4, \ldots u_{2k-1}u_{2k}\}$ et $M_2 = \{u_2u_3, u_4u_5, \ldots u_{2k-2}u_{2k-1}, u_{2k}u_1\}$.
- Comme $w(M_1) + w(M_2) = \text{OPT}$, on conclut que $w(M_1) \leq \text{OPT}/2$ ou $w(M_2) \leq \text{OPT}/2$.
- En particulier, $w(M) \leq \text{OPT}/2$.



Le facteur d'approx. de Double-Tree ne peut pas être amélioré



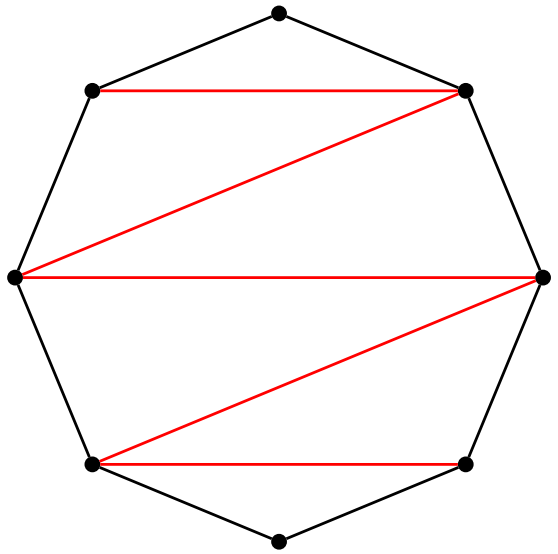
- Soit W_n la "roue" à $n+1$ sommets, avec n impair et toutes les arêtes de poids 1.
- Manifestement, OPT $= n+1$
- Soit T l'étoile dont le centre est le moyeu de W_n.
- Supposons que le cycle eulérien de G est de la forme

  $v_0, v_1, v_0, v_3, v_0, v_5, \ldots$
- Alors, on obtient le cycle

  $v_0, v_1, v_3, \ldots, v_n, v_2, v_4, \ldots, v_{n-1}, v_0$

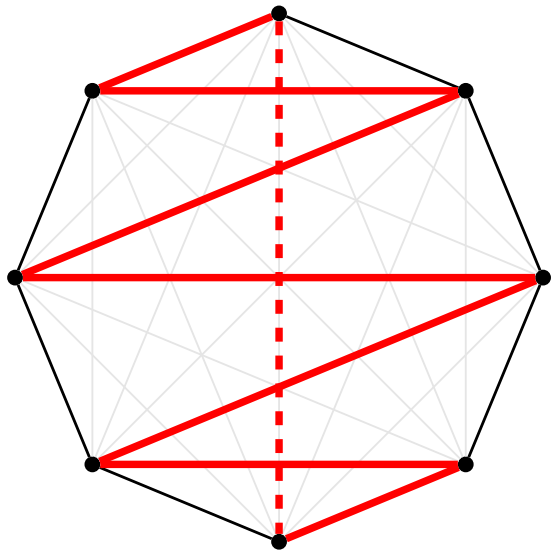
  de poids $2n$.

Le facteur d'approx. de Christofides ne peut pas être amélioré (1/2)



- Soit H le graphe qui consiste de :
  - un cycle C_n de longueur n, pour n paire, avec sommets $v_1, v_2, \ldots, v_n$
  - une chaîne P en forme de zigzag entre v_1 et $v_{n/2+1}$

Le facteur d'approx. de Christofides ne peut pas être amélioré (2/2)



- Soit G le graphe complet avec ensemble de sommets $V(G) = V(H)$, avec pondération $w_{uv} = \mathrm{dist}_H(u, v)$ pour chaque arête uv.
- Manifestement, OPT $= n$
- J consiste de l'arête $v_1 v_{n/2+1}$, de poids $n/2$.
- Le graphe qui en résulte est un cycle hamiltonien de G de poids $n - 1 + n/2 = 3n/2 - 1$.
- Cela tend vers $3n/2$ lorsque $n \to \infty$.