Implémentation d'algorithmes de recherche opérationnelle. Projet phase 1

Polytechnique Montréal

MTH6412B

Xavier Lebeuf et Geoffroy Leconte

Septembre 2021

```
md"""
# Implémentation d'algorithmes de recherche opérationnelle. Projet phase 1

Polytechnique Montréal

MTH6412B

Xavier Lebeuf et Geoffroy Leconte

Septembre 2021
"""
```

1. Importation des fichiers

```
md"""
# 1. Importation des fichiers
"""
```

Main.workspace2.plot_graph

```
begin
    using Plots
    using PlutoUI

    include("projet/phase1/edge.jl")
    include("projet/phase1/node.jl")
    include("projet/phase1/graph.jl")
    include("projet/phase1/read_stsp.jl")
end
```

2. Type Edge

Nous avons créé un type Edge, très similaire au type déjà existant Node. Le type Edge comporte trois attributs: le nom, un tuple contenant les deux extrémités et le poids.

```
md"""
# 2. Type Edge

Nous avons créé un type Edge, très similaire au type déjà existant Node. Le type Edge comporte trois attributs: le nom, un tuple contenant les deux extrémités et le poids.
"""
```

3 et 4. Étendre les fonctions existantes

Nous avons ajouté un attribut vecteur contenant des types Edge au type Graph. Nous avons aussi ajouté un paramètre I pour ce nouvel attribut. Une seconde boucle for à été ajoutée à la fonction `show` pour montrer les arêtes également.

```
md"""
# 3 et 4. Étendre les fonctions existantes

Nous avons ajouté un attribut vecteur contenant des types Edge au type Graph. Nous avons aussi ajouté un paramètre I pour ce nouvel attribut. Une seconde boucle for à été ajoutée à la fonction `show` pour montrer les arêtes également.
"""
```

5. Prendre en compte les poids

La fonction `read_edges` contient maintenant un second tableau auquel on ajoute les poids des arêtes à chaque itération. Nous avons aussi ajouté une condition `if` de façon à ne pas ajouter de poids et d'arêtes si ces dernières ont le même noeud comme successeur et prédécesseur. Ces arêtes ne sont pas utiles dans le contexte du voyageur de commerce et elles alourdissent l'algorithme.

```
md"""
# 5. Prendre en compte les poids

La fonction `read_edges` contient maintenant un second tableau auquel on ajoute les poids des arêtes à chaque itération. Nous avons aussi ajouté une condition `if` de façon à ne pas ajouter de poids et d'arêtes si ces dernières ont le même noeud comme successeur et prédécesseur. Ces arêtes ne sont pas utiles dans le contexte  du voyageur de commerce et elles alourdissent l'algorithme.
"""
```

6. Fonction créant un graphe

Voici la fonction créant un objet de type Graphe à partir d'un fichier .tsp, ainsi qu'un exemple avec le fichier bayg29.tsp. Le code montré ci-dessous est le même que celui contenu dans le fichier mainphase1.

```
md"""
# 6. Fonction créant un graphe

Voici la fonction créant un objet de type Graphe à partir d'un fichier .tsp, ainsi qu'un exemple avec le fichier bayg29.tsp. Le code montré ci-dessous est le même que celui contenu dans le fichier mainphase1.
"""
```

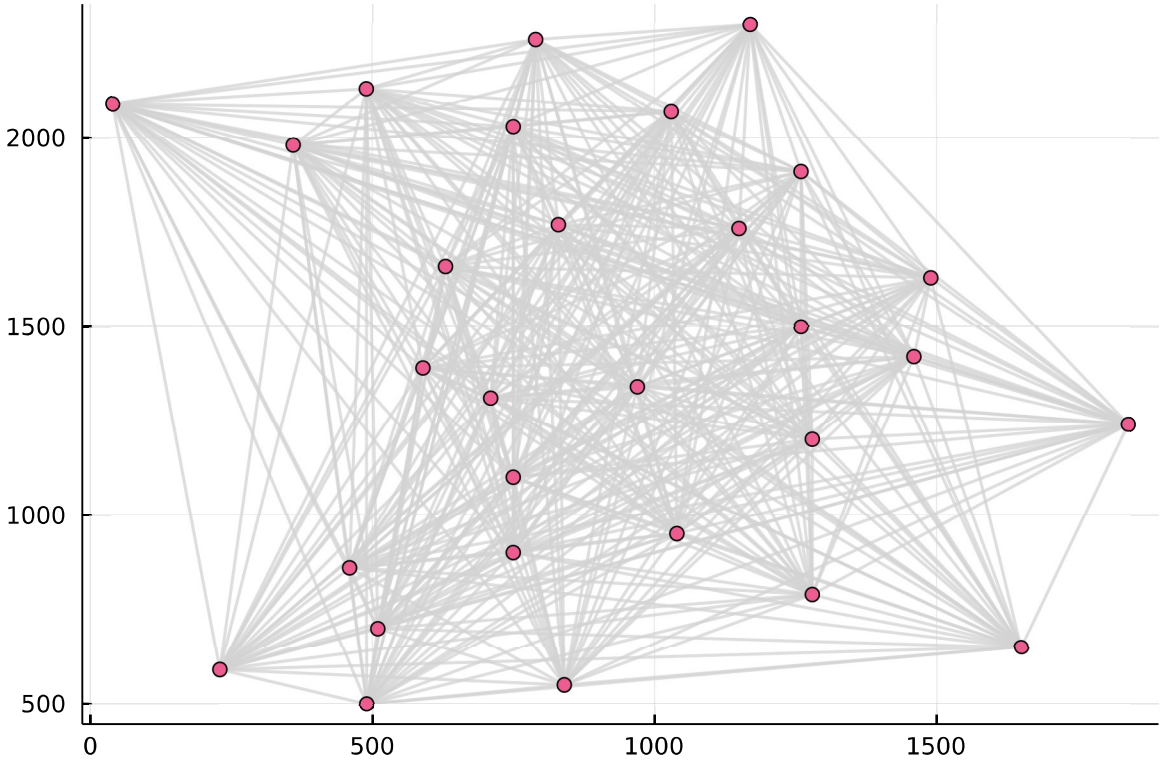
Main.workspace2.createGraph

```
"""
Renvoie un objet de type Graphe à partir d'un fichier .tsp
"""
function createGraph(graphname::String, filename::String)

    dict = read_header(filename)
    edge_list, weight_list = read_edges(dict, filename)
    node_list = read_nodes(dict, filename)

    G = Graph(graphname, Node{Vector{Float64}}[], Edge{Int64}[])

    for no in node_list
        newnode = Node(string(no[1]), no[2])
        add_node!(G, newnode)
    end

    for i in 1:length(edge_list)
        newedge = Edge(string(edge_list[i][1], "↔", edge_list[i][2]), edge_list[i], weight_list[i])
        add_edge!(G, newedge)
    end
    G
end
```

```
G = createGraph("bays29", raw"./instances/stsp/bays29.tsp");
```

```
Graph bays29 has 29 nodes and 812 edges.
Node 5, data: [750.0, 2030.0]
Node 16, data: [1280.0, 1200.0]
Node 20, data: [590.0, 1390.0]
Node 12, data: [1170.0, 2300.0]
Node 24, data: [1260.0, 1500.0]
Node 28, data: [1260.0, 1910.0]
Node 8, data: [1490.0, 1630.0]
Node 17, data: [230.0, 590.0]
Node 1, data: [1150.0, 1760.0]
Node 19, data: [1040.0, 950.0]
Node 22, data: [490.0, 500.0]
Node 23, data: [1840.0, 1240.0]
Node 6, data: [1030.0, 2070.0]
Node 11, data: [840.0, 550.0]
Node 9, data: [790.0, 2260.0]
Node 14, data: [510.0, 700.0]
```

```
with_terminal() do
show(G)
end
```




```
plot_graph(raw"./instances/stsp/bayg29.tsp")
```

7. Branche phase1 sur GitHub

Lien: https://github.com/XavierLebeuf/mth6412b-starter-code/tree/phase1

```
md"""
# 7. Branche phase1 sur GitHub

Lien: https://github.com/XavierLebeuf/mth6412b-starter-code/tree/phase1
"""
```