Replanification ferroviaire en temps réel par résolution hybride IA/RO

Léo Baty

SNCF Direction Innovation & Recherche, département MEV, groupe MOD

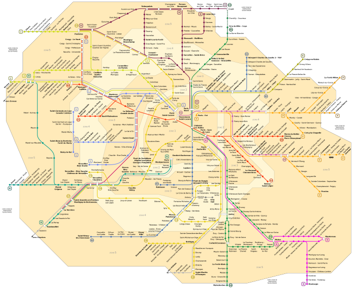
Ecole Nationale des Ponts et Chaussées, MPRO

Sous la direction de Hugo Belhomme

Logo MPRO

Contexte industriel : Transilien

- Service de transport en commun le plus utilisé en Île-de-France.
- Hybride entre le métro et le train : fréquence élevée proche de Paris, fréquence faible loin de Paris.
- Contexte temps réel



Plan de transport : solution du problème de planification

Graphe $G = (\mathcal{E}, \mathcal{A}, w)$ orienté pondéré.

Noeuds (évènements)

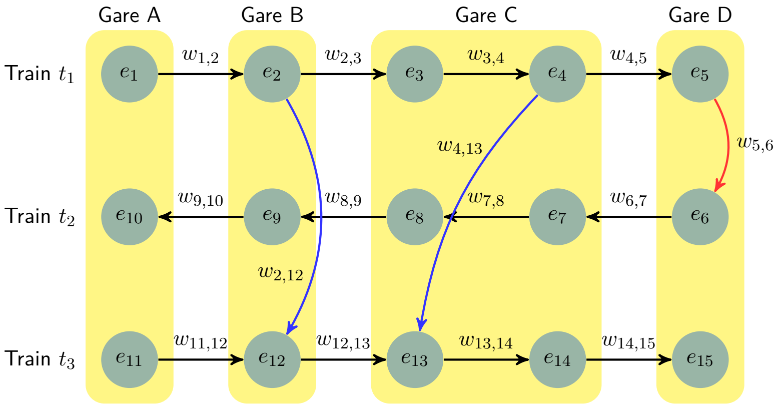
Soit un évènement $e \in \mathcal{E}$.

- **Type d'évènement** : terminus origine, terminus destination, arrivée, départ, ou passage.
- **Point remarquable** : lieu.
- **Train** associé.
- **Horaire théorique** : heure prévue.

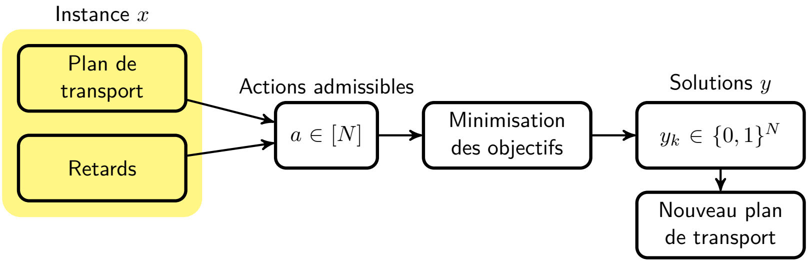
Types d'arc

- Arc de **parcours**
- Arc de **retournement**
- Arc d'**espacement**

Exemple de plan de transport



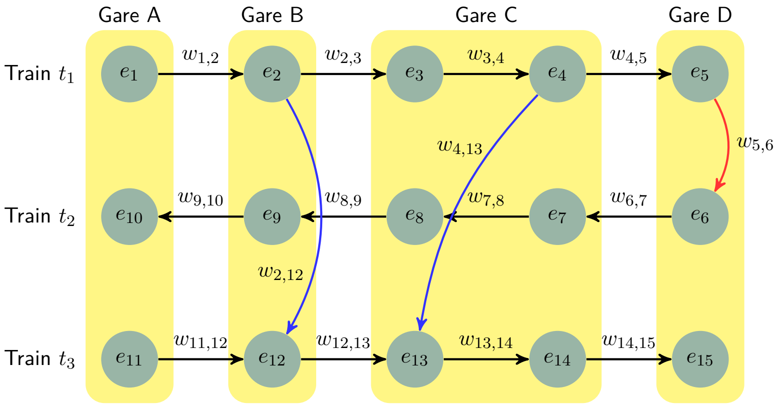
Le problème de replanification ferroviaire



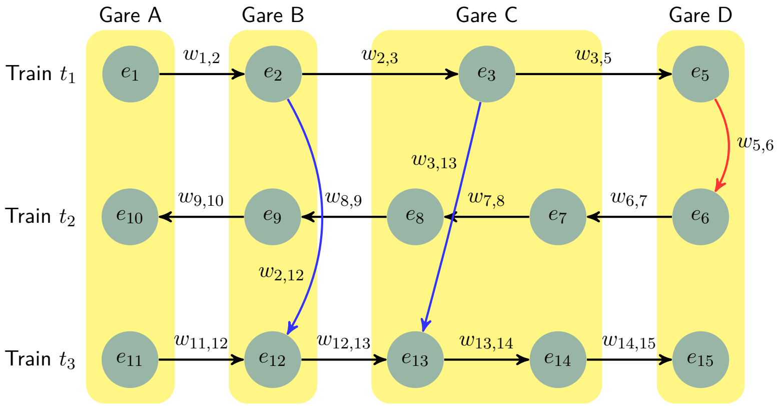
Objectifs à minimiser :

1. **Objectif voyageur** : temps d'attente + temps de voyage.
2. **Durée de la perturbation** : temps qu'il faut au réseau pour absorber les retards actuels.
3. **Nombre d'actions de régulation**.
4. **Retard des engins** : somme des retards de chaque engin.

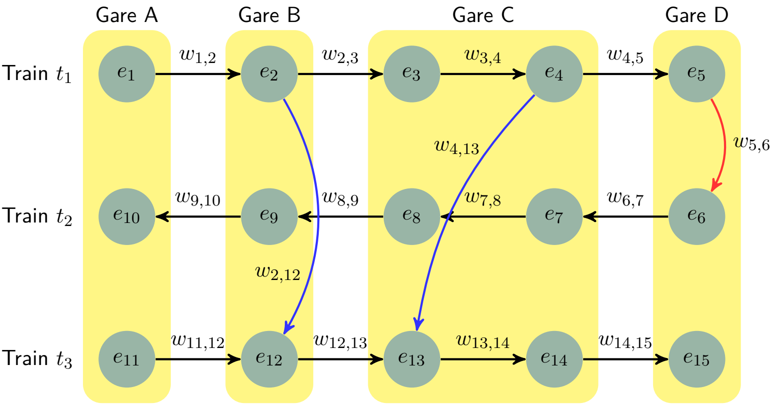
Exemple


Gare A
Gare B
Gare C
Gare D
Train t_1
Train t_2
Train t_3
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
$w_{1,2}$
$w_{2,3}$
$w_{3,4}$
$w_{4,5}$
$w_{5,6}$
$w_{6,7}$
$w_{7,8}$
$w_{8,9}$
$w_{9,10}$
$w_{4,13}$
$w_{2,12}$
$w_{11,12}$
$w_{12,13}$
$w_{13,14}$
$w_{14,15}$

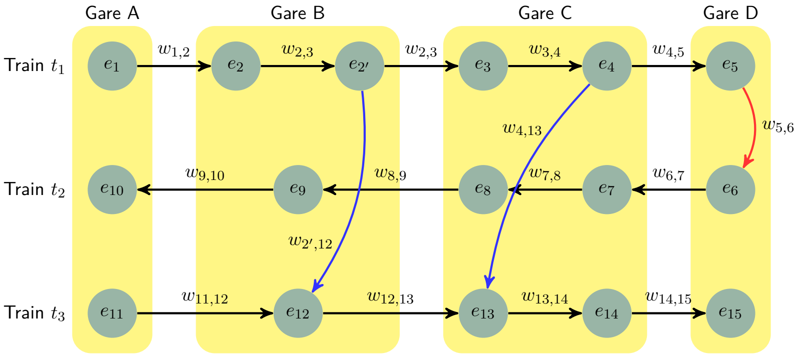
Exemple : suppression d'arrêt



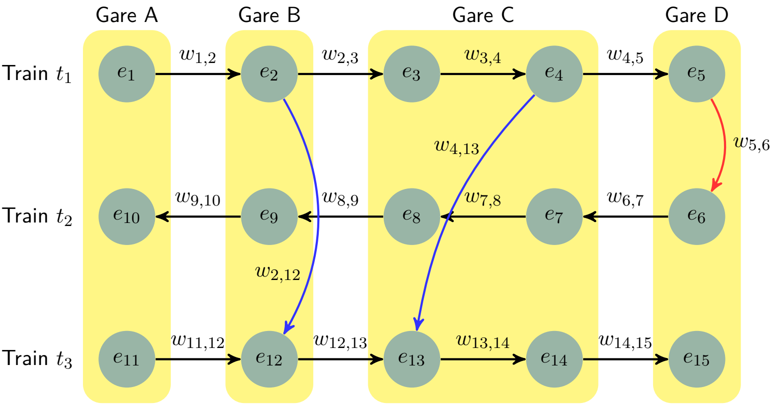
Exemple : ajout d'arrêt



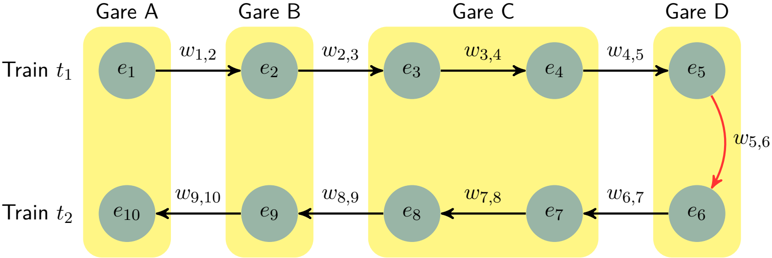
Exemple : ajout d'arrêt



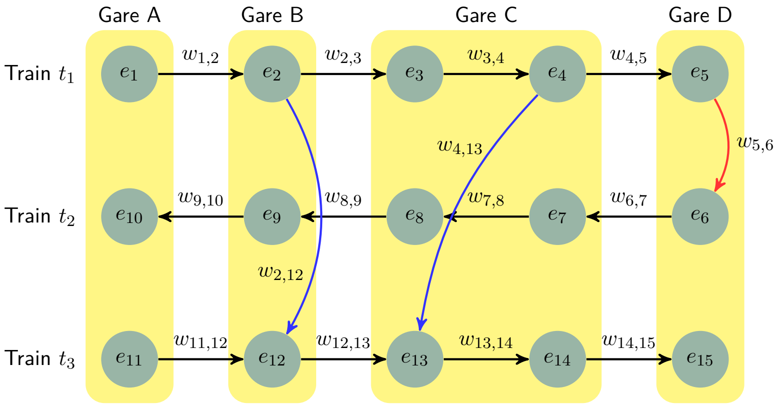
Exemple : suppression de train



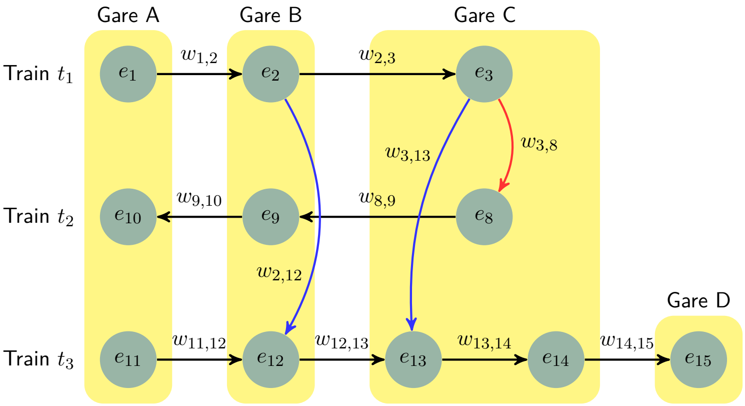
Exemple : suppression de train



Exemple : limitation



Exemple : limitation



L'outil actuel

- Gestionnaires du plan de Transport et de l'Information voyageur (GTI)

- Outil d'aide à la décision en cours de développement, deux algorithmes d'optimisation :

  1. **Recherche exhaustive** : parcours des 2^N solutions
     $\Rightarrow$ trop long en temps réel
  2. **Heuristique add** (3 branches) : algorithme glouton
     $\Rightarrow$ utilisé en pratique

Plan

Pipeline

Objectif : mise en place d'approches d'apprentissage automatique dans le but d'enrichir les algorithmes d'optimisation de l'outil. Utilisation des jeux de données d'historiques.

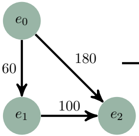
$$\boxed{\text{Instance } x} \xrightarrow{f_\theta} \underset{\text{Scores}}{\boxed{(s_a)_{a\in[N]} = f_\theta(x)}} \xrightarrow{\mathcal{A}} \underset{\text{Solutions}}{\boxed{y = \mathcal{A}(f_\theta(x))}}$$

- f_θ : réseau de neurones, $\theta \in \mathbb{R}^d$ paramètres à apprendre
- $\mathcal{A}$: heuristique guidée par les scores

Réseaux de neurones sur des graphes

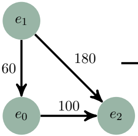
Problèmes potentiels des réseaux de neurones classiques (matrice d'adjacence) avec des graphes :

- Information sparse
- Entrées de grande dimension
- Nombre de noeuds variables
- Non-invariance par permutation des sommets :



Matrice d'adjacence

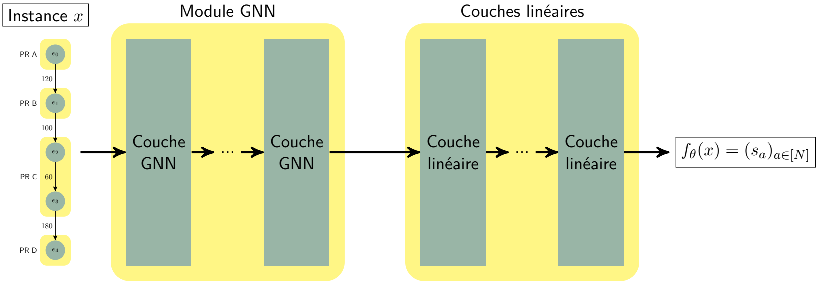
$$\longrightarrow \begin{pmatrix} 0 & 60 & 180 \\ 0 & 0 & 100 \\ 0 & 0 & 0 \end{pmatrix}$$



Matrice d'adjacence

$$\longrightarrow \begin{pmatrix} 0 & 0 & 100 \\ 60 & 0 & 180 \\ 0 & 0 & 0 \end{pmatrix}$$

Graph Neural Network (GNN)



Entrée

$$\begin{pmatrix} \begin{array}{r|ccc|c} -10 & 1 & 0 & 0 & 0 \\ 50 & 0 & 0 & 1 & 1 \\ 300 & 0 & 1 & 0 & 1 \\ 400 & 0 & 0 & 0 & 0 \\ 600 & 0 & 0 & 0 & 0 \end{array} \end{pmatrix} \longrightarrow$$

Node embeddings

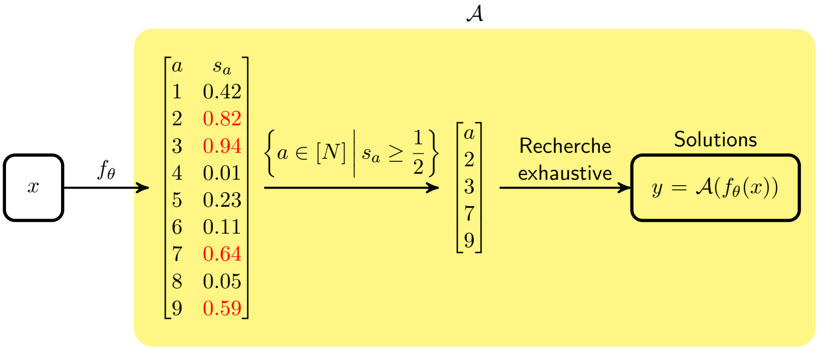
$$\begin{pmatrix} \begin{array}{rr|rr} 0.0178 & 0.0379 & -0.3394 & -0.272 \\ -0.0764 & -0.0468 & -0.3394 & -0.272 \\ -0.1054 & -0.1011 & -0.3394 & -0.272 \\ -0.0727 & -0.0897 & -0.3394 & -0.272 \\ -1.4602 & -1.1603 & -0.3394 & -0.272 \end{array} \end{pmatrix} \longrightarrow$$

Prédiction

$$\begin{pmatrix} 0.4 & 0.45 & 0.72 \\ 0.01 & 0.43 & 0.85 \\ 0.23 & 0.1 & 0.16 \\ 0.78 & 0.08 & 0.59 \\ 0.87 & 0.27 & 0.63 \end{pmatrix}$$

Heuristique : recherche exhaustive guidée par GNN

Parcours de tous les scénarios composés uniquement d'actions ayant un scores élevé.



$\Rightarrow$ moins de scénarios parcourus, temps d'éxécution plus rapide.

Génération d'instances

Données d'historiques

- Plans de transport depuis mai 2018.
- Pour chaque jour et train : parcours théorique, et heures de passages observées.

Construction d'un jeu d'instances

- Ligne L
- Uniquement des fenêtres de 2018 - 2019.
- 10 fenêtres par jour de durée 1h.

Création d'un jeu de données d'apprentissage

Labélisation des graphes

- On résout chaque instance en *offline* avec la recherche exhaustive $\Rightarrow$ front de Pareto exact.
- Pour chaque action admissible a, on définit un label c_a :

$$c_a = \mathbb{1}_{\{a \text{ appartient à au moins une solution du front de Pareto}\}}$$

Dataset résultant

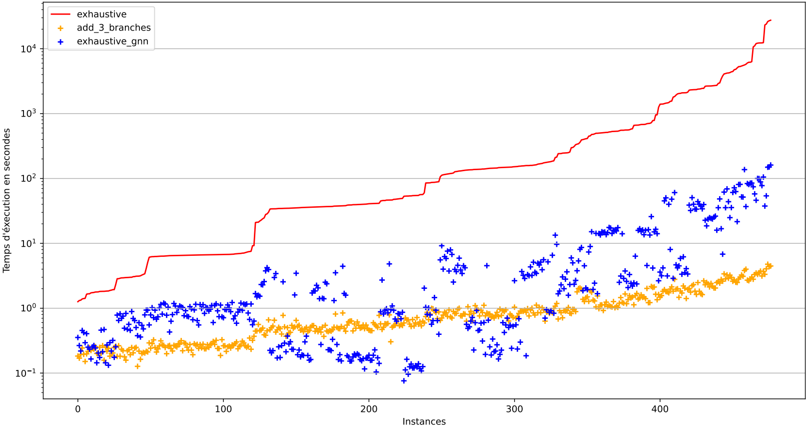
- Training/validation/test : 900/100/500 graphes.
- En moyenne, 1000 noeuds et 1500 arcs.
- En moyenne, 140 actions admissibles, et 7 actions admissibles positives (i.e. telles que $c_a = 1$).

Apprentissage

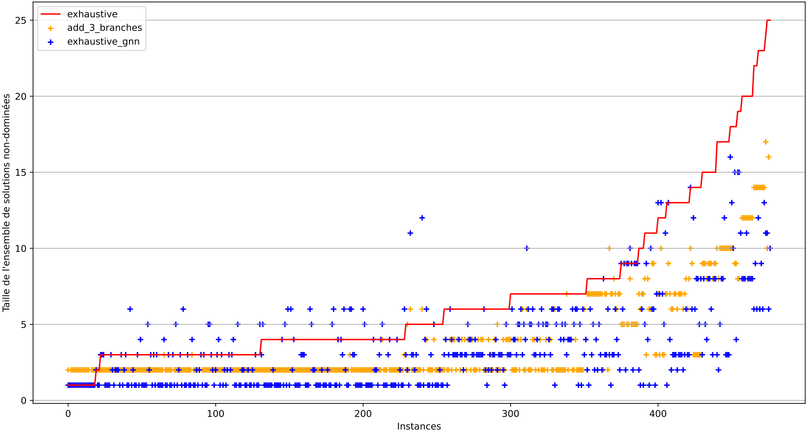
- Implémentation en Python avec `PyTorch Geometric`.
- Exécution sur un serveur distant : 6-12h sur 10 cpu pour chaque modèle.
- Loss : variante de la binary-cross-entropy

Modèle	**Loss**	**Précision négative**	**Précision positive**
Meilleur modèle	2.47	0.85	0.82

Temps d'exécution



Nombre de solutions non-dominées



Bilan

Objectif	Objectif voyageur	Retard des engins	Durée de la perturbation
Recherche exhaustive	3917.8	6774.9	1205.7
Add 3 branches	4025.4	6829	1269.1
Exhaustive GNN	3991.2	6791.7	1264.9

Conclusion

Contributions

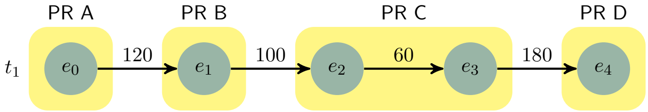
1. Génération d'instances labélisées à partir des données d'historique.
2. Utilisation des instances pour entraîner un modèle de Graph Neural Network.
3. Heuristique utilisant le modèle GNN.

Perspectives

- Apprentissage structuré et Fenchel-Young loss.
- Amélioration de $\mathcal{A}$: métaheuristiques multi-objectifs.
- Utilisation d'autres jeux de données.
- Apprentissage par renforcement avec GNN.

Merci pour votre attention

Encodage des graphes : exemple



Features d'entrée

$$\begin{pmatrix} -10 & 1 & 0 & 0 & 0 \\ 50 & 0 & 0 & 1 & 1 \\ 300 & 0 & 1 & 0 & 1 \\ 400 & 0 & 0 & 0 & 0 \\ 600 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Arcs

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

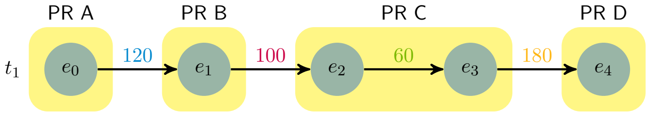
Poids

$$\begin{pmatrix} 120 & 100 & 60 & 180 \end{pmatrix}$$

Labels

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & 0.5 \\ -1 & 1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Encodage des graphes : exemple



Features d'entrée

$$\begin{pmatrix} -10 & 1 & 0 & 0 & 0 \\ 50 & 0 & 0 & 1 & 1 \\ 300 & 0 & 1 & 0 & 1 \\ 400 & 0 & 0 & 0 & 0 \\ 600 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Arcs

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

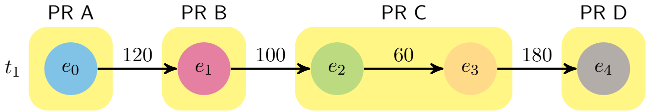
Poids

$$\begin{pmatrix} 120 & 100 & 60 & 180 \end{pmatrix}$$

Labels

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & 0.5 \\ -1 & 1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Encodage des graphes : exemple



Features d'entrée

$$\left(\begin{array}{c|ccc|c} -10 & 1 & 0 & 0 & 0 \\ 50 & 0 & 0 & 1 & 1 \\ 300 & 0 & 1 & 0 & 1 \\ 400 & 0 & 0 & 0 & 0 \\ 600 & 0 & 0 & 0 & 0 \end{array}\right)$$

Arcs

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

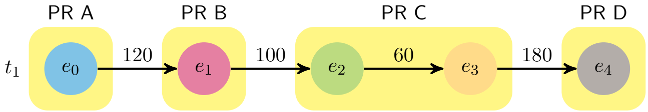
Poids

$$\begin{pmatrix} 120 & 100 & 60 & 180 \end{pmatrix}$$

Labels

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & 0.5 \\ -1 & 1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Encodage des graphes : exemple



Features d'entrée

$$\left(\begin{array}{c|ccc|c} -10 & 1 & 0 & 0 & 0 \\ 50 & 0 & 0 & 1 & 1 \\ 300 & 0 & 1 & 0 & 1 \\ 400 & 0 & 0 & 0 & 0 \\ 600 & 0 & 0 & 0 & 0 \end{array}\right)$$

Arcs

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Poids

$$\begin{pmatrix} 120 & 100 & 60 & 180 \end{pmatrix}$$

Labels

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & 0.5 \\ -1 & 1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Module GNN : notations

Soit $G = (\mathcal{E}, \mathcal{A}, w)$ en entrée du module GNN.

- K le nombre de couches du GNN
- d : nombre de features par sommet
- $\mathbf{X} \in \mathbb{R}^{|\mathcal{E}| \times d}$: **features** sur chacun des sommets (x_e ses lignes).
- $\mathbf{h}_e^{(k)}$: vecteur de sortie de la couche $k \in [K]$, appelé **hidden embedding**
- $\mathbf{z}_e$: vecteur en sortie du module GNN, appelé **node embedding**.

Module GNN : les maths

1. Pour chaque noeud, initialisation :

$$\boxed{\mathbf{h}_e^{(0)} = \mathbf{x}_e, \ \forall e \in \mathcal{E}}$$

2. Itération $k \leq K-1$: mise à jour de l'**hidden embedding** :

$$\boxed{\mathbf{h}_e^{(k)} = \mathbf{ReLU}\left(\mathbf{\Theta}_1 \mathbf{h}_e^{(k-1)} + \mathbf{\Theta}_2 \sum_{f \in \mathcal{N}(e)} w_{f,e} \mathbf{h}_f^{(k-1)}\right), \ \forall e \in \mathcal{E}}$$

$\mathbf{\Theta}_1$ et $\mathbf{\Theta}_2$ deux matrices de poids de taille $d \times d$, et $\mathbf{ReLU}(x) = \max(x, 0), \ \forall x \in \mathbb{R}$.

3. Sortie de la dernière couche :

$$\boxed{\mathbf{z}_e = \mathbf{h}_e^{(K)} = \mathbf{\Theta}_1 \mathbf{h}_e^{(K-1)} + \mathbf{\Theta}_2 \sum_{f \in \mathcal{N}(e)} w_{f,e} \mathbf{h}_f^{(K-1)}, \ \forall e \in \mathcal{E}}$$

Concaténation du graph embedding

Le node embedding z_e pour chaque contient une information locale sur son voisinage.

Graph embedding z_G

$$z_G = \frac{1}{card(\mathcal{E})} \sum_{e \in \mathcal{E}} z_e$$

Entrée du module de loss :

$$[z_e, z_G], \forall e \in \mathcal{E}$$

Loss : notations

- s_e : sortie (vecteur de taille 3) de la dernière couche linéaire pour le noeud e.
- s_a : **sortie** (scalaire) associée à ;l'action a (récupéré sur s_e avec e associé à a).
- y_a : **score** de l'action a.
- On dit qu'une action admissible est **positive** si elle appartient à au moins une action du front, i.e. $y_a > 0$.
  On note n^+ le nombre d'actions admissibles positives.
- Sinon, on dit qu'elle est **négative** ($y_a = 0$).
  On note n^- le nombre d'actions admissibles positives.

Module de loss : fonction de loss

$$\forall x \in \mathbb{R},\ \sigma(x) = \frac{1}{1+\exp(-x)} \in [0,1]$$

Loss de classification

$$\frac{1}{n} \sum_{a \text{ adm.}} -\left(\mathbb{1}_{y_a>0} \log \sigma(s_a) + (1-\mathbb{1}_{y_a>0}) \log \sigma(1-s_a)\right)$$

Loss de classification séparée

$$p_1 \left(\frac{1}{n^-} \sum_{\substack{a \text{ adm.} \\ \text{t.q. } y_a=0}} -\log \sigma(1-s_a) \right) + p_2 \left(\frac{1}{n^+} \sum_{\substack{a \text{ adm.} \\ \text{t.q. } y_a>0}} -\log \sigma(s_a) \right)$$

Métriques de performances : précisions

Précision positive (resp. négative)

Proportion d'actions admissibles *positives* (resp. *négatives*) qui sont bien classifiées par le GNN.

Exemple (actions admissibles en rouge) :

Labels : $\mathbf{y}$

$$\begin{pmatrix} 0 & -1 & -1 \\ -1 & -1 & 0.5 \\ -1 & 1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

$\mathbb{1}_{\mathbf{y}>0}$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Prédiction : $\sigma(\mathbf{s})$

$$\begin{pmatrix} 0.4 & 0.45 & 0.72 \\ 0.01 & 0.43 & 0.85 \\ 0.23 & 0.1 & 0.16 \\ 0.78 & 0.08 & 0.59 \\ 0.87 & 0.27 & 0.63 \end{pmatrix}$$

Précision positive : 0.5

Métriques de performances : rang du dernier positif

Rang d'une action a

Classement de a si l'on ordonne les actions par prédictions $\sigma(s_a)$ décroissantes. Noté r_a.

Rang du dernier positif

Rang le plus élevé parmi les actions admissibles positives :

$$\max_{a \text{ adm.}} r_a$$

Exemple (actions admissibles positives en vert) :

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \color{green}{0.94} & \color{green}{0.82} & 0.64 & \color{green}{0.59} & 0.42 & 0.23 & \color{green}{0.11} & 0.05 & 0.01 \end{bmatrix}$$

Rang du dernier positif : 7

Métriques de performances : distance linéaire moyenne

Distance linéaire moyenne

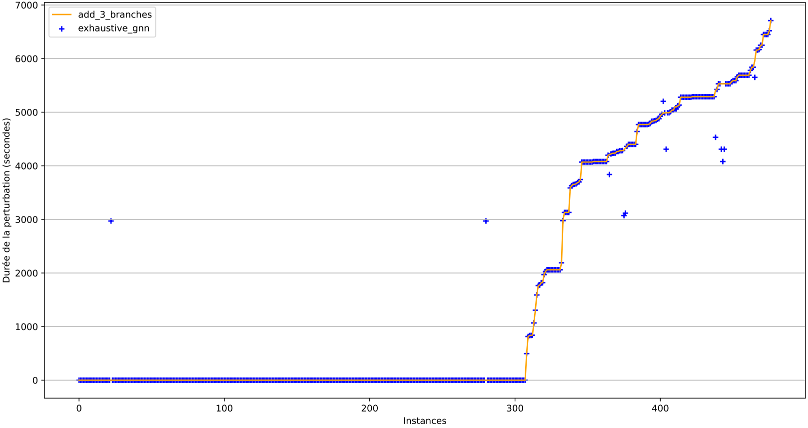
$$\text{DistanceMoyenneLinéaire(G)} = \frac{1}{n^+} \sum_{a \text{ admissible}} \max(n^+ - r_a, 0)$$

Exemple (actions amissibles positives en vert) :

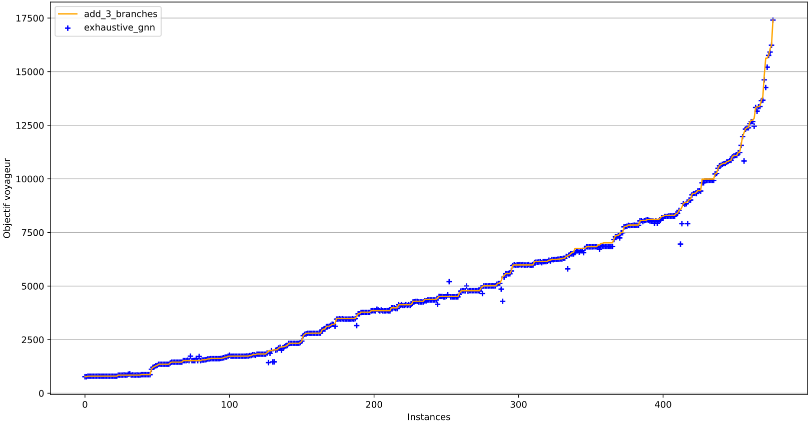
$$\left[\begin{array}{cccc|ccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \color{green}{0.94} & 0.82 & \color{green}{0.64} & 0.59 & \color{green}{0.42} & 0.23 & \color{green}{0.11} & 0.05 & 0.01 \end{array}\right]$$

Distance linéaire moyenne : $\dfrac{0+0+1+3}{4} = 1$

Objectifs : durée de la perturbation



Objectifs : objectif voyageur



Objectifs : Retard des engins