Killian
LE GOFF

Rapport de la solution proposée au problème de réseau de portes

Ce rapport explique la démarche utilisée pour chercher à résoudre le problème posé. Une attention particulière a été portée à la définition des termes utilisés (traduction des exigences pratiques vers une expression mathématiques) et à la mise en place d'une méthodologie claire.

1) Modéliser le problème

L'objectif de cette première partie est de réfléchir à la modélisation du problème, en vue de générer des données. La représentation doit être exploitable à la fois par l'humain et par l'algorithme. Une première intuition est de travailler avec des graphes (grâce à la librairie ***graphviz*** sous python et des éléments de théorie des graphes).

A partir d'une **matrice d'adjacence** (le coefficient en ligne i et colonne j représente le nombre porte(s) qui permettant de rentrer dans la piece j en venant de la pièce i) qui représente un réseau de portes, on peut modéliser sous forme de graphe un réseau de portes. Les **noeuds** du graphe correspondent aux différentes pièces, et les **arcs orientés** correspondent aux portes entre les pièces (la flêche indiquant le sens extérieur => intérieur, qu'on appellera **sens direct**). On donne un exemple ci-dessous à partir d'un plan de maison (fig.1).

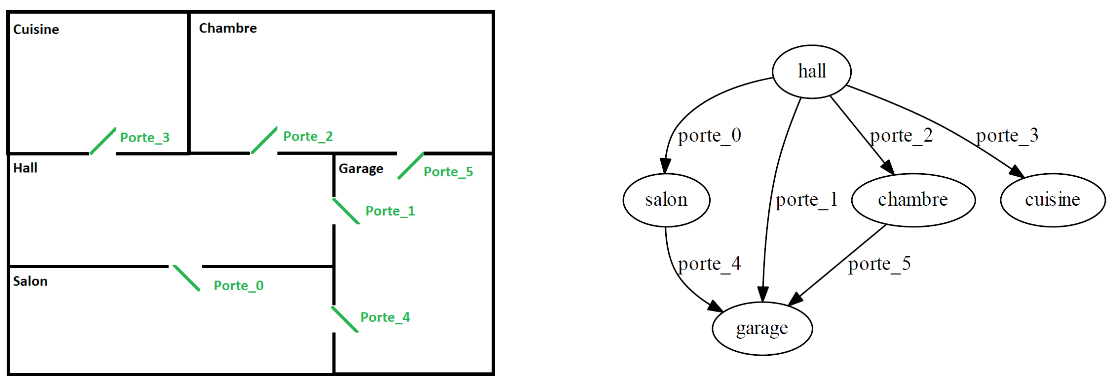

Fig. 1: Représentation d'un réseau de portes à partir d'un plan

Pour obtenir un **réseau de porte cohérent** (càd qui peut correspondre à une situation réelle, sans incohérences qui pourraient fausser l'interprétation), il faut adresser plusieurs problèmes. Ce qui caractérise un réseau de porte cohérent sont les conditions suivantes :

- Toutes les pieces ont au moins une porte = chaque noeud est relié à au moins un autre noeud
- Il est toujours possible de suivre une chemin de portes dans le sens de sortie afin de sortir.
- Toutes les portes qui relient deux pièces sont dans le même sens (pas de contradiction entre plusieurs portes).

Pour définir un réseau de porte, il faut définir le réseau de pièces sous-jacent. Pour vérifier la deuxième condition de cohérence, on propose de définir deux types de pièces :

- **intérieures** : les pièces dans le réseau, dont l'accès est régulé par les badges

– **extérieures** : les pièces hors du réseau, mais connectées au réseau intérieur.
Les personnes considérées comme hors du réseau de portes seront considérées dans ces pièces extérieures (en pratique, ces pièces peuvent être abstraites).

Les détails de l'élaboration de l'algorithme pour créer un réseau de portes se trouvent en commentaire du script fourni en annexe. La **fonction "creer_matrice_adjacence"** permet de créer une matrice d'adjacence en précisant le nombre de pièces total, de pièces extérieures et le nombre de portes souhaité (à cause de l'étape de correction de la matrice, cf ci-dessous). Elle procède en plusieurs étapes :

-Création d'un porte pour chacune des pièces, vers une pièce aléatoire, avec une condition sur le sens si on travaille avec un pièce extérieure.

-Création de portes supplémentaires afin d'atteindre le nombre de portes requis, toujours en étudiant l'éventualité de traiter une piece extérieure.

-Correction de la matrice obtenue afin qu'elle soit cohérente (cf. Paragraphe ci-dessus et les commentaires du script pour les détails).

Cette fonction nous permet ainsi de générer une matrice d'adjacence aléatoire et cohérente, dont on peut choisir le paramétrage, et qu'on pourra représenter de la façon suivante en utilisant la bibliothèque graphviz ('**g.view()**'). Un exemple avec les paramères suivant :

N_porte = 12 ;
N_piece = 10
N_piece_ext = 2

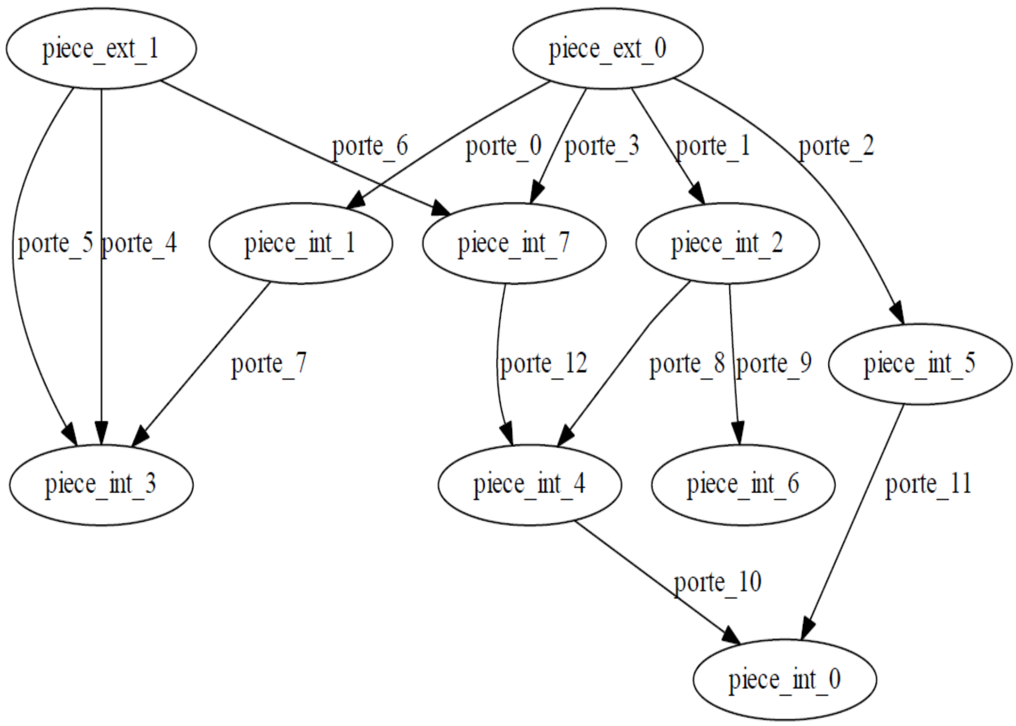

Fig. 2 : Exemple de réseau de portes généré par matrice d'adjacence

2) Création d'un jeu de données

L'objectif de cette deuxième partie est de déterminer comment générer des événements sur le réseau de porte ainsi généré. On souhaite modéliser le déplacement de plusieurs individus dans en introduisant une notion de probabilité de sortir/entrer dans une pièce ou d'y rester. Une première idée consiste à utiliser la théorie des **chaînes de Markov**, en utilisant un temps en valeurs discrètes.

On définit la **notation** suivante :

T(0) - *instant initial de l'étude*
delta_T - *intervalle entre les différents dates*
T(n) = T(0) + n * delta_T – *n-ième date de l'étude (correspond à l'état après le déplacement n)*

X(i) - *nombre de personnes présentes dans la pièce i*
P(i,j) – *probabilité qu'un individu passe de la pièce i à la pièce j*

Si la matrice P est constante, on a la relation suivante :

Pour tous i et n, $\mathbf{X(i,T(n+1)) = \sum_j P(i,j) * X(j,T(n))}$

Les coefficients P(i,j) forment une matrice qu'on appelle **matrice de transition**. Dans un sytème concret, on peut supposer que cette matrice évolue continuellement mais qu'il est possible d'en avoir une bonne estimation si on a assez de données. Pour conserver une part de variabilité dans les données générées, on va introduire des **variations** dans les coefficients de cette matrice.

Dans le réseau de portes généré, il est probable que plusieurs portes fassent le lien entre deux mêmes pièces. Dans ce cas, le **coefficient P(i,j)** est la somme des probabilité des portes. Pour simplifier le paramétrage du problème, on peut supposer que pour les portes qui font passer de i à j sont équiproba

Si la matrice P est la matrice identité, les personnes ne se déplaceraient jamais. On fixe la sous-matrice composée des N_piece_ext colonnes et lignes à une matrice diagonale, afin que les personnes ne se déplacent pas entre les différentes pièces extérieures au système (ce qui traduit l'hypothèse que les personnes n'utilisent que le même accès pour rentrer dans le réseau de pièces). Les N_piece_ext premières colonnes de la matrice correspondent à la probabilité que les individus sortent du réseau de porte, et les N_pieces_ext premières colonnes donnent la probabilité que les personnes y rentrent. Cette délimitation donne 4 sous-matrices qui correspondent aux **cas d'utilisation du réseau** de portes, comme illustré sur la figure 3.

Pour simuler les différents **comportements** qu'on peut espérer avoir dans une journée, on utilisera 4 matrices de transition différentes qui seront utilisées de façon cyclique pour mettre à jour l'état du système en fonction de l'heure de la journée :

-La **nuit** (20h-8h): personne ne rentre dans le réseau
-Le **matin** (8-10h) : les personnes rentrent uniquement dans le réseau
-La **journée** (10h-16h): les personnes bougent un peu dans le réseau
-Le **soir** (16-20h): les personnes sortent du réseau

La fonction "**creer_matrice_transition**" permet de créer une matrice de transition cohérente à partir d'un **vecteur caractérisant les comportements** des personnes à une date donnée (voir le script pour les détails).

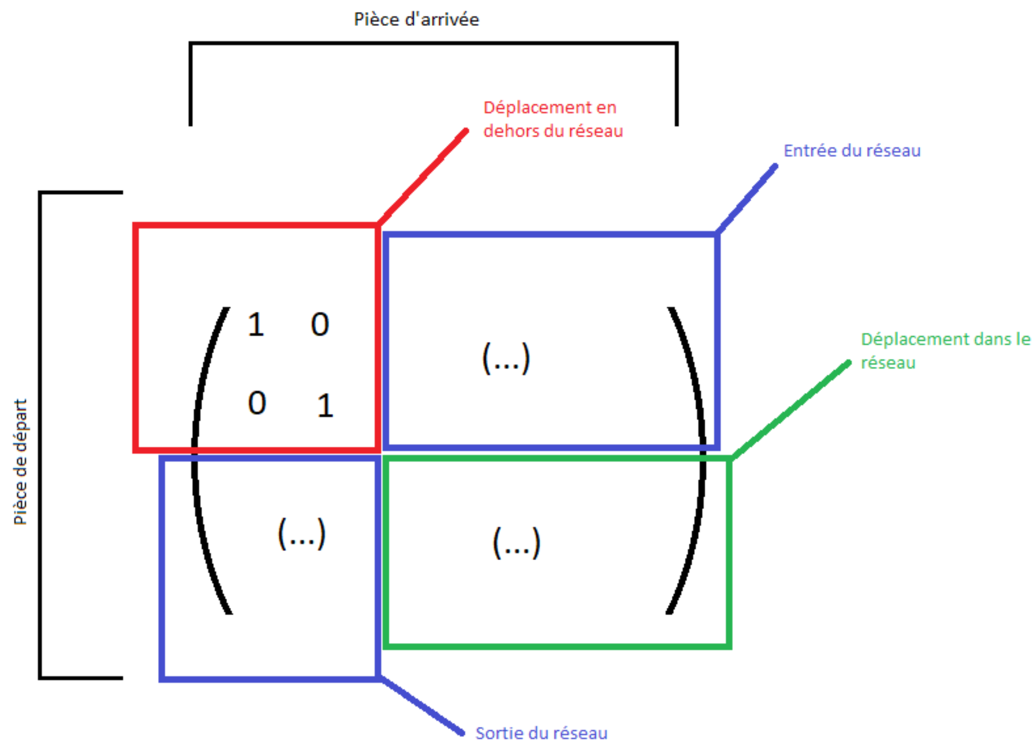

Fig.3 Cas d'utilisation du réseau et matrice de transition (exemple avec 2 pièces extérieures)

Avec ces fonctions, on est capables de modéliser le déplacement d'un certain nombre de personnes dans le réseau, en choisissant :

- Paramètres du problème : N_piece, N_piece_ext, N_porte (type = int),
- Paramètres de la simulation : N_personne (type = int) ; delta_t (type = timedelta) ; t0, t_final (type = datetime), coeff_comportement (type = list, éléments de type = int ou float)

Pour générer des données on utilise le jeu de paramètres suivant :

N_piece = 120 ; N_piece_ext = 30 ; N_porte = 200

N_personne = 40 ; delta_t = 10 secondes, t0 = 01/01/2019 à 00:00:00, t_final = 08/01/2019 à 00:00:00

coeff_comportement =
[1,60,40,5] le matin (8-10h)
[1,100,30,5] la journée (10-16h)
[100,5,1,100] le soir (16-20h)
[1000,1,1,100] la nuit (20-8h)
[100000,1,1,1] le weekend

3) Analyse du jeu de données

L'objectif de cette troisième partie est de définir différentes métriques afin d'étudier les données dont on dispose. On regardera les données selon différents axes, qui sont précisés ci-dessous.

D'une part, on étudie le **nombre d'utilisation des portes** sur l'ensemble de la période d'acquisition des données. Pour celà on met en place les 3 premières métriques suivantes :

métrique 1 - la porte la plus utilisée quelque soit le sens
métrique 2 - la porte avec le plus d'entrée
métrique 3 - la porte avec le plus de sorties

(en pratique ces métriques seront des listes, s'il y a égalité entre plusieurs portes)

D'autre part, on envisage d'étudier l'évolution nombre d'utilisation des portes du réseau dans le temps. Pour ce faire, on transforme les données sous-forme de série temporelle, et on en extrait les métriques suivantes :

metrique 4 – date à laquelle il y a le plus de passage de portes, quel que soit le sens
metrique 5 – date à laquelle il y a le plus de passage de portes dans le sens direct
metrique 6 – date à laquelle il y a le plus de passage de portes dans le sens indirect

metrique 7 – date à laquelle il y a le moins de passage de portes, quel que soit le sens
metrique 8 – date à laquelle il y a le moins de passage de portes dans le sens direct
metrique 9 – date à laquelle il y a le moins de passage de portes dans le sens indirect

A ces métriques, on ajoute un **graphique du trafic** sur le réseau qui comprend le nombre de portes ouvertes dans le sens entrant (en bleu), dans le sens sortant(en vert) et dans les deux sens (en rouge), et qui sera sauvegardé dans un fichier "**graphique_activite.png**". Le graphique obtenu pour le jeu de données simulé avec les paramètres du paragraphe précédent est donné en figure 4.

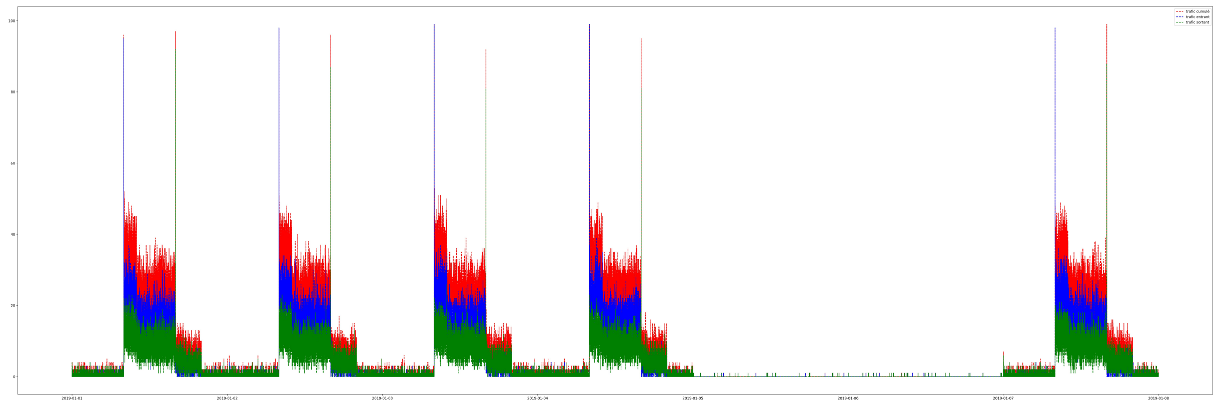

Fig.4 Graphique de l'activité sur le réseau

Sur un vrai jeu de données, ce graphique permet d'identifier des intervalles de temps sur lesquels les comportements sont similaires. On observe ici, une très faible activité le week-end, et le reste de la semaine, quatre plages horaires distinctes (matin, journée, soir, nuit). Pour comprendre les comportements des utilisateurs, on trace également l'évolution du nombre de personnes dans le réseau en fonction du temps, qui sera sauvegardé dans le fichier "**graphique_effectif.png**", qui donne pour l'exemple généré :

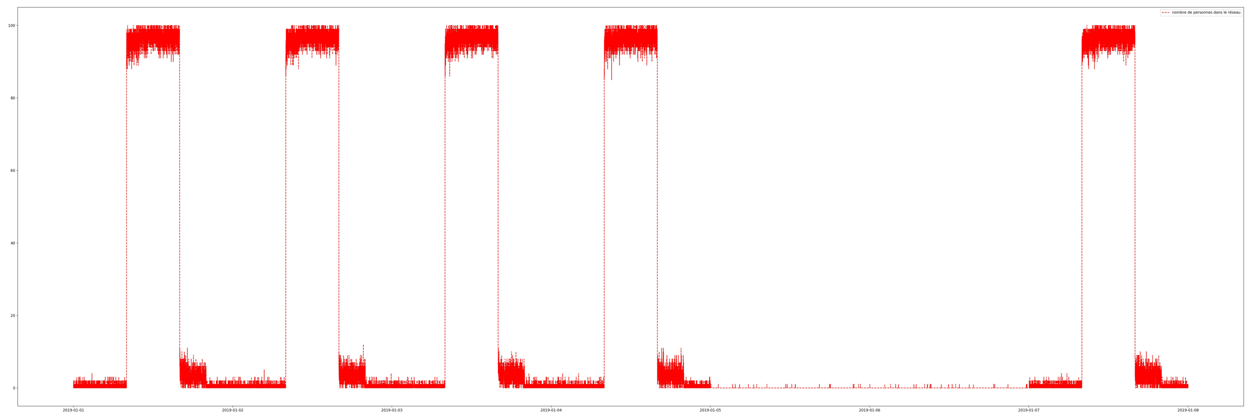

Fig.5 Evolution de l'effectif du réseau dans le temps

L'exploitation de ces deux graphiques nous permet de retrouver les comportements qu'on a souhaité modéliser, à savoir :

- Le week-end le trafic est très faible, les employés ne sont pas sur leur lieu de travail en général.
- Le matin le trafic est très important, les employés cherchent à aller à leurs postes, mais n'empruntent pas toujours le même chemin et font parfois des détours, sans forcément sortir du réseau.
- La journée, les employés peuvent emprunter les portes dans les deux sens mais peu sortent du réseau, ce qui illustre la mobilité dans la journée entre services par exemple.
- Le soir, l'effectif diminue brusquement, et le employés qui passent des portes le font majoritairement dans le but de sortir du réseau.
- La nuit, il y a beaucoup moins d'activité sur le réseau, ce qui traduit par exemple un effectif de nuit réduit par rapport à celui de jour.

4) Conclusion et perspectives

Le travail réalisé a permis de simuler un réseau de portes, et de générer des données aléatoires suivant des lois de comportements qu'on a précisées. Sur un cas pratique, le réseau de porte sera connu mais la matrice de transition sera inconnue, et si on la détermine sur les différentes plages horaires identifiés grâce à l'analyse des données, on sera capables de prédire les effets de l'évolution du réseau (par exemple l'ajout/le retrait d'une porte, etc...) sur les déplacements et de prédire le trafic. Un point d'honneur a été porté à la modélisation du problème, afin d'obtenir des données cohérentes, et qui se rapprochent de celles qu'on peut espérer obtenir sur un problème réel.

Il serait intéressant d'étudier des données sur un problème concret, et de voir dans quelle mesure la modélisation par un processus de markov est pertinente, et de déterminer le nombre de matrices de transitions qu'il faudrait estimer pour avoir une bonne modélisation du trafic.