

École des Ponts

ParisTech

# Modélisation de l'interface quai-train

**Manal AKHANOUSS, Joy-Rose DUNOYER DE SEGONZAC, Benoit DUVAL,  
Ramzi SAYAH**

sous la direction de Mostepha Khouadjia (IRT SystemX) et Eric Duceau (directeur scientifique du département IMI)

**RÉSUMÉ :** Des millions de personnes utilisent les transports en commun ferroviaires de Paris et sa banlieue. Certaines gares connaissent de fortes densités de personnes, au niveau du quai ou dans les trains. Cette densité peut augmenter si il y a des retards sur la ligne, entraînant un flux d'échange entre le train et le quai de plus en plus grand. La SNCF souhaite modéliser l'interface quai-train afin de prédire les temps d'arrêt du train et donc les éventuels retards supplémentaires. On utilise ici le logiciel NetLogo pour réaliser des simulations multi-agents. Comme le temps d'échange au niveau du train correspond au temps d'échange le plus long sur toutes les portes, nous avons effectué nos simulations sur une seule porte, que nous appelons porte critique. Le résultat principal d'une simulation est le temps au bout duquel tous les usagers souhaitant monter ou descendre ont accompli leur objectif. D'après les observations sur le terrain, le temps d'échange reste acceptable jusqu'à un certain niveau de remplissage des voitures. Notre objectif est de modéliser cet échange et de mettre en évidence des paramètres et caractéristiques qui influent sur ce temps d'échange. Nous avons créé plusieurs scénarios pour étudier l'échange. Un premier scénario de montée où des personnes sur le quai veulent monter dans un train vide. Certaines personnes "polies" essayent de garder une distance avec leur voisin et des personnes "impolies" n'hésitent pas à coller les autres pour avancer. Un deuxième scénario de descente où des personnes dans la voiture veulent descendre et personne ne veut monter. Ces personnes doivent éviter des obstacles, dont nous étudions l'impact. Un troisième scénario avec des personnes qui veulent descendre, monter ou rester sur place. Chaque personne a un coefficient de diffusion de respect et de politesse. Lorsque deux personnes veulent aller au même endroit, ce sera la personne qui diffuse le plus de respect et la moins polie qui aura la priorité.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Modélisation des sous problèmes</b>	<b>4</b>
2.1	Montée . . . . .	4
2.2	Placement . . . . .	4
2.3	Descente . . . . .	5
2.4	Échange . . . . .	6
<b>3</b>	<b>Outils de simulation et résultats</b>	<b>6</b>
3.1	Fonctionnement de Netlogo . . . . .	6
3.2	Paramètres des agents et des simulations . . . . .	6
3.3	Scénario de montée . . . . .	7
3.4	Descente . . . . .	14
3.5	Échange . . . . .	19
<b>4</b>	<b>Recul sur les résultats et comparaison au cas réel</b>	<b>23</b>
<b>5</b>	<b>Conclusion</b>	<b>24</b>
<b>6</b>	<b>Bibliographie</b>	<b>26</b>

# 1 Introduction

L'utilisation de plus en plus importante des transports en commun et en particulier des trains et métros pousse les acteurs du réseau ferroviaire à optimiser toutes les étapes du voyage, de l'assignation des trains jusqu'au dimensionnement des trains et des quais. Le problème de l'échange des voyageurs à l'interface quai-train est primordial car permet de calculer les temps d'arrêt du train et donc de prédire au mieux les éventuelles propagations de retard. Le problème se posant pour des trains fréquents voire très fréquents, l'échange doit se faire de manière rapide. Il sera alors question ici de modélisations d'interface quai-train pour des lignes de type francilien du réseau SNCF.

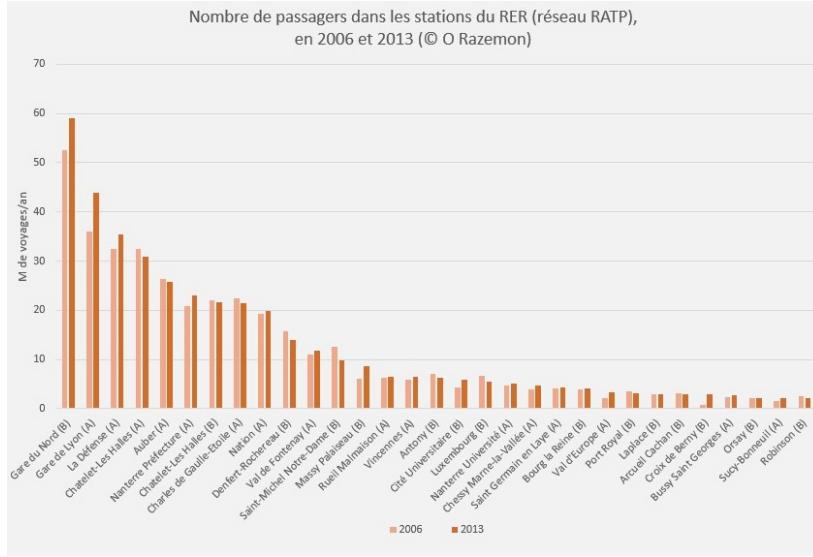


FIGURE 1 – Nombre de passagers par an (Olivier Razemon)

La modélisation des temps d'échange est d'autant plus importante que la gare est fréquentée, le modèle sera ainsi très utile pour les gares parisiennes qu'on peut voir dans la figure 1 issue d'un article de *Le Monde* [4].

L'objectif est de repérer les seuils à partir desquels les temps d'arrêt explosent en terme de durée. Pour modéliser cela, nous avons utilisé le simulateur multi-agents libre d'accès NetLogo[6]. Nous faisons le choix d'utiliser une méthode par agents plutôt qu'une méthode particulière pour la modélisation des comportements, la méthode particulière est plus facile à implémenter dans le cas de scénarios de montée dans un train vide ou de sortie de train (compression/décompression). Mais dans nos simulations, la *politesse* va jouer un rôle, or il n'existe pas de particule polie qui vont attendre que d'autres vont passer avant de se déplacer, en se mettant sur le côté des portes lors de l'ouverture par exemple.

Ainsi notre modèle, par agents, se limite à un train exclusivement en deux dimensions, on considère qu'il n'y a pas d'escaliers à l'intérieur du train. De plus, on ne s'intéresse dans notre modèle qu'à une seule porte d'une des rames du train, en effet les retards causés par les échanges proviennent exclusivement de ce qu'il se passe au niveau de la porte la plus problématique. Plusieurs choix ont été réalisés quant au comportement de déplacement des agents, cette section sera abordée plus tard en détails et est cruciale car la politesse des acteurs peut avoir un très grand impact sur les temps d'échange et donc d'arrêt. Nous verrons plus loin que le problème, complexe, d'un échange de passagers peut être décomposé en plusieurs sous problèmes

relativement indépendants. Chacun de ces sous problèmes présente une solution qui nous permet de vérifier le modèle en le confrontant à divers scénarios.

Une fois le modèle calibré, satisfaisant les différents scénarios de test prévus à cet effet et dont les résultats sont proches de ceux que l'on peut trouver dans les données fournies par la SNCF, nous ferons varier les paramètres du problème afin d'identifier les variables les plus importantes à prendre en compte lors d'une modélisation d'échange à l'interface quai-train. Ces paramètres peuvent être de l'ordre mécanique ou physique comme l'agencement de la rame mais également de l'ordre social et psychologique sur l'influence de la stratégie de chaque agent ou groupe d'agents.

## 2 Modélisation des sous problèmes

Comme dit ci-dessus, nous pouvons diviser le problème en quatre parties indépendantes, nous verrons les objectifs de chaque sous problème et quelles en sont les variables importantes.

### 2.1 Montée

On s'intéresse à la montée de personnes, initialement sur le quai, dans une rame de train de dimensions finies. Les résultats qui nous intéressent sont :

- Le temps de montée de tous les passagers ( $+\infty$  si impossible de faire monter tout le monde)
- la densité de personnes dans la rame au cours du temps

L'objectif est ici de pouvoir modéliser le phénomène de **compression**, étant donné que l'objectif de chaque passager est de monter à bord de la rame et qu'on considère ici que tout le monde joue d'un point de vue collectif (chacun cherche à minimiser son espace pour pouvoir faire monter le plus de monde). On cherchera un point de rupture du temps d'échange en fonction du nombre de personnes sur le quai, cette fonction serait linéaire dans le cas nominal (peu de personnes) puis deviendrait strictement concave, tendant vers  $+\infty$  pour le nombre maximal de personnes pouvant entrer.

Cette notion de comportement collectif sera évidemment remise en cause après, notamment dans la phase d'échange.

On peut identifier les paramètres importants de notre problème :

- Dimensions de la rame
- Nombre de personnes cherchant à monter
- Largeur de la porte
- Distance minimale d'éloignement entre les passagers

### 2.2 Placement

Le problème du placement est lié aux trois autres sous problèmes. Le placement des personnes sur le quai, cherchant à monter à bord de la rame, influence :

- La montée (si les personnes sont en file indienne il n'y a pas de phénomène de bouchons au niveau de la porte)
- La descente, car ces personnes représentent des obstacles pour les gens cherchant à partir du quai
- L'échange, car ces personnes peuvent bloquer les portes de sortie par leur placement

Le placement des usagers influence également ce qu'on appelle la largeur réelle de la porte : si les personnes sur le quai se rapprochent trop de la porte ils limitent la largeur de sortie des personnes à l'intérieur du train.

Dans l'article de S.Seriani et T.Fujiyama[5], les résultats des expériences montrent que la distribution initiale des passagers londoniens présents dans une station de métro sont en "V-inversé" par rapport à la porte, voire en arc-de-cercle lorsque la densité de personnes est trop importante 2. Cette notion de V-inversé est également caractéristique des passagers de la ligne 14 du métro parisien à l'arrêt *Bibliothèque François Mitterrand* dans l'article de F.Elleuch [1].



Figure 6 – Example of “arch” behaviour of passengers waiting to board at Westminster station

FIGURE 2 – Comportement en arc de cercle, image issue de [5]

Néanmoins dans ces deux articles, l'emplacement des portes est connu par les utilisateurs même lorsque le train n'est pas présent (présence de barrières de sécurité révélant l'emplacement des portes). Toutefois on peut considérer qu'entre l'arrivée du train et l'ouverture des portes, la grande majorité des agents ont prêt l'information sur la position de la porte et ont pu se déplacer en conséquence.

## 2.3 Descente

Dans ce scénario, des personnes initialement dans le train descendent sur un quai où personne ne cherche à monter. À contrario de la montée, on regarde ici un phénomène de **décompression** les paramètres sont donc les mêmes que la montée. L'objectif est d'identifier l'influence d'obstacles "vivants" sur la descente, par exemple la présence de personnes, dans le train, devant les portes mais qui ne descendant pas : ce phénomène est récurrent dans les réseaux de transport francilien. Ainsi on s'intéresse

- Au temps de descente de tous les usagers
- A la densité de personnes au cours du temps, en particulier autour des dits obstacles
- influence de la position des personnes ne descendant pas (assises, non assises, devant les portes, etc...)

## 2.4 Échange

L'échange est le coeur de notre problème, là où seule la descente ou la montée peuvent prendre du temps à cause de phénomène de compression/décompression l'échange fait également intervenir des comportements sociaux plus ou moins individuels ou collectif. Dans l'étude de l'échange on s'intéresse donc exclusivement à ce qui se passe autour de la porte. Nous souhaitons modéliser assez finement le comportement des passagers sans pour autant faire une analyse sociale des mouvements d'échange. C'est pourquoi nous représentons dans nos modèles un indice de priorité, où une personne est plus ou moins prioritaire à faire son mouvement suivant (avancer pour la majorité des cas). Nous utilisons également un indice d'impolitesse. Dans un cadre idéal, tous les usagers respectent l'ordre naturel de l'échange :

- chaque personne voulant descendre, descend sans se bousculer (le plus proche de la porte part en premier)
- les personnes souhaitant monter attendent que tous ceux voulant descendre du train soient descendus
- les usagers montent sans se bousculer

La réalité n'est pas aussi simple, certains descendant en poussant certains passagers, d'autres montent beaucoup trop tôt créant ainsi un phénomène de blocage. Cet indice d'impolitesse, dont on donnera les valeurs et les bornes plus loin, indique alors la propension de chaque agent à faire une action alors qu'il n'est pas prioritaire.

## 3 Outils de simulation et résultats

### 3.1 Fonctionnement de Netlogo

Nous utilisons pour nos modélisations le logiciel NetLogo[6]. C'est un simulateur multi-agents où on retrouve le concept de

- tortues/agents : les usagers ici, qui vont se déplacer.
- ticks : qui sont une unité de mesure du temps : si une même simulation est réalisée sur deux ordinateurs différents le nombre de ticks sera le même (mais pas forcément le temps de simulation qui dépend des paramètres de la machine)
- patchs : qui nous servent à représenter l'espace (en deux dimensions ici) où se déplacent les tortues.

Un programme Netlogo se déroule généralement ainsi : il y a une première phase de setup où le décor est initialisé. Les tortues sont placées dans ce décor et on leur donne des caractéristiques (par exemple un objectif : monter, descendre, rester sur place).

Ensuite, on demande aux tortues d'effectuer une action (avancer vers leur objectif, éviter les autres personnes, laisser passer...) pendant un certain nombre de ticks, ou jusqu'à ce qu'une condition soit respectée (toutes les tortues sont arrivées à destination par exemple).

### 3.2 Paramètres des agents et des simulations

Les paramètres de nos agents et leurs domaines de définitions respectifs sont

Tous ces paramètres permettent à la fois de décrire entièrement l'agent (sa position, son objectif et son état) mais également ce qui va relever du comportement social, une impolitesse élevée indique que l'agent

Paramètres de l'agent	Domaine de définition
Position x, y	$\mathbb{R}^2$
Objectif O	Montée, Descente, Rien
Etat actuel e	Quai, Train
Impolitesse i	[0,1]
Diffusion du respect d	[0,1]

va se considérer comme prioritaire pour chaque action quitte à gêner les autres. La propension à diffuser le respect, quant à elle, indique le fait d'être naturellement prioritaire, c'est le cas dans la vraie vie des femmes enceintes, des personnes à mobilité réduite, des personnes âgées etc.

Pour les simulations, les paramètres sont intuitifs :

Paramètres d'une simulation	Domaine de définition
Dimensions du train $L, l$	$\mathbb{R}^2$
Largeur de la porte $w$	$\mathbb{R}$
Répartition initiale des personnes sur le quai $\rho$	Densité de probabilité sur $\mathbb{R}^2$
Probabilité pour un agent de descendre $p_{desc}$	[0,1]
Probabilité pour un agent de monter $p_{mont}$	[0,1] (supposé =1)
Nombre de personnes à quai $n_{quai}$	$\mathbb{R}^+$
Nombre de personnes dans la rame $n_{rame}$	$\mathbb{R}^+$
Ratio de personnes malpolies	[0,1]

On fera varier ces paramètres afin de tester les limites de notre modèle dans les différents scénarios proposés ci-dessus. Pour l'automatisation des simulations nous utilisons PyNetLogo [2] qui nous permet de faire la jonction entre NetLogo et Python.

### 3.3 Scénario de montée

Dans ce scénario où il y a uniquement des personnes qui montent. Un effet qu'il était nécessaire d'avoir est celui de compressibilité. Par cela, on entend que les personnes doivent, si elles le peuvent garder le plus d'espace possible avec les autres personnes, tout en atteignant leur objectif : monter dans le train. Ainsi, on doit abandonner l'hypothèse des autres scénarios où il y avait une personne par patch.

Disons qu'une personne fait la taille de 3 patchs. On représente une personne par un cercle de diamètre 40 cm, comme cela a déjà été fait dans la thèse de B. Kabalan [3]. Un patch mesure donc  $\frac{0.40}{3} = 0.13$  m. Une voiture mesure environ 21 m de long et 3m de large. Une voiture classique possède 3 portes. Cela dépend des modèles de trains, mais nous nous sommes basés sur ces chiffres. Nous simulerons une porte, l'espace voiture qui lui est réservé est un rectangle de 7m de longueur par 3m de largeur. En réalité, les personnes peuvent circuler dans la voiture, et parfois entre les voitures, mais nous nous sommes concentrés sur la porte critique, celle où il y a le plus de personnes, donc nous n'avons pas pris cela en compte dans nos hypothèses.

Cela nous donne une longueur de train  $L = \frac{21 \times 3}{0.40} = 52$  patchs. Et la largeur  $l = \frac{3 \times 3}{0.40} = 22$  patchs. La porte mesure 2m, donc  $w = 15$  patchs de largeurs.

On centre la porte, et pour délimiter les zones dans le train, on ajoute des murs, ce qui nous donne 3 zones dans le train, devant la porte, à gauche et à droite.

	Taille en mètres	Taille en patchs
Longueur de la voiture	7	52
Largeur de la voiture	3	22
Largeur du quai	4	30
Longueur de la porte	2	15
Largeur du couloir	1.07	8
Diamètre d'une tortue	0.40	3

On peut afficher le cadre que cela nous donne, cf figure 3. Les patchs roses sont la voiture, les patchs bleus le quai, en vert la porte et en noir, les murs qui sont impossible à traverser.

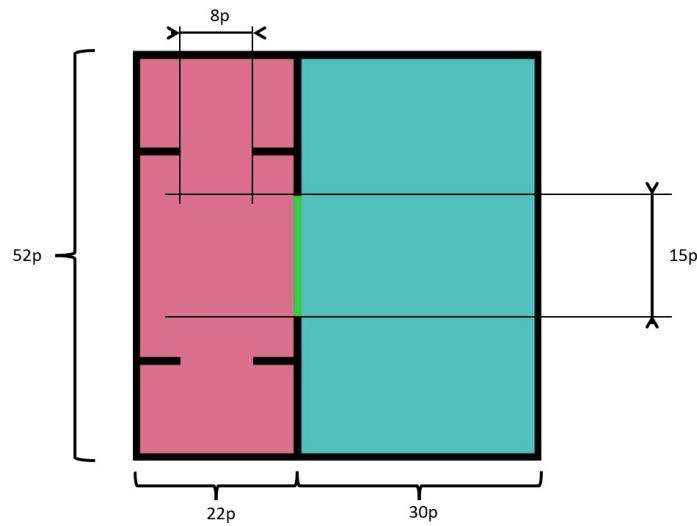


FIGURE 3 – Cadre dans le scénario de montée avec compression

On fait ensuite apparaître un certain nombre de tortues sur le quai, qui veulent monter. La distribution des tortues est uniforme dans un premier temps.

Ensuite, nous devons définir le comportement des tortues.

Une tortue est sur le quai, elle veut aller dans la voiture. Pour cela, elle aura comme objectif le patch de la porte le plus proche d'elle. Il y a ensuite une distinction de cas, selon si la personne est "polie" ou non.

Si elle est polie, elle essaiera d'avancer sur un de ses quatre patchs voisins uniquement si elle ne "colle" aucune autre tortue, c'est à dire qu'il doit y avoir au moins 1 patch d'écart entre elle et une autre tortue. Elle ira sur le patch accessible qui la rapproche le plus de son objectif, à savoir la porte.

Si la personne n'est pas polie, elle cherchera à se rapprocher de son objectif, la porte, en se déplaçant vers le patch voisins accessible qui la rapproche le plus, et n'hésitera pas à coller les autres tortues pour passer.

Si l'on fait uniquement cela, on observe un phénomène qui n'est pas naturel : les tortues vont s'amaser sur le côté des portes et laisser un couloir de passage au milieu, alors qu'il n'y a personne à descendre.

C'est normal car les patchs sur les côtés de la porte ont plus de chance d'être les patchs objectifs des tortues, vu que la distribution sur le quai est uniforme, cf figure 4

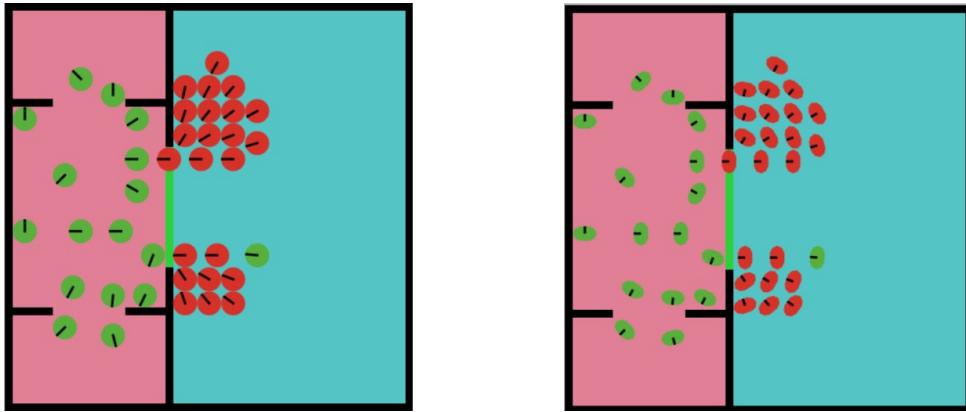


FIGURE 4 – Problème couloir libre

Les deux images sont les mêmes. La plus réaliste au modèle est celle avec les cercles, mais pour un affichage plus graphique, les personnes peuvent être représentées comme des ovales. Le trait noir indique la direction où les tortues regardent.

Ainsi, on fait en sorte que si les tortues attendent loin de la porte, sur le côté et que le couloir au centre est dégagé, elles s'y engagent, ce qui est plus naturel.

Une fois les tortues arrivées sur la porte, elles veulent rentrer dans le train. Elles vont estimer le nombre de personnes actuellement dans la voiture et vont en déduire un rayon libre maximum qu'ils peuvent espérer avoir. Elles vont ensuite se diriger vers un patch qui respecte cette condition.

Si elles sont parmi les premiers à monter, elles ne vont pas rester juste devant la porte.

Tant que les personnes n'ont pas atteint leur place objectif, elles se déplacent. Or cette place objectif varie au cours du temps car les autres tortues se déplacent, ce qui fait que les tortues bougent tout le temps, ce n'est pas réaliste, mais si on force les tortues à s'arrêter au bout d'un moment, le train se bloque et les gens ne peuvent plus rentrer.

On ajoute aussi le fait que si une première tortue colle une deuxième tortue, cette deuxième tortue va essayer d'avancer dans la direction opposée. Ie, si une tortue vient me coller à gauche, j'essaye d'aller à droite, etc.

On observe des blocages dans le cas où toutes les tortues sont polies. À cause de l'arc de cercle qui se crée autour de la porte, aucune tortue n'ose avancer car elle se trouve trop proche d'autres tortues, et donc la situation est bloquée. Nous voulons éviter cela, ainsi on rajoute un paramètre de *patience* aux tortues polies. Au bout de *patience* ticks passés sans bouger sur le même patch, les tortues polies deviennent impolies, elles n'hésitent plus à se coller à d'autres tortues et cela permet de débloquer la situation.

Les simulations se terminent lorsque toutes les personnes sont montées dans le train.

Les paramètres dont nous disposons sont le *nombre de personnes qui montent*, la *proportion de personnes polies au début de la simulation*, la *patience* des personnes polies. On pourrait aussi modifier la

largeur du quai, la largeur des portes, l'espace libre autour de soi désiré par les personnes polies ou impolies.

Pour voir la réalisation de cela, on peut voir sur la figure 5 avec 20 personnes polies avec une patience de 20 ticks qui veulent monter et sur la figure 6 avec 20 personnes impolies, en partant de la même situation initiale que dans la situation précédente. Les tortues en vert ne collent aucune autre tortue, alors que celles en rouge collent d'autres tortues.

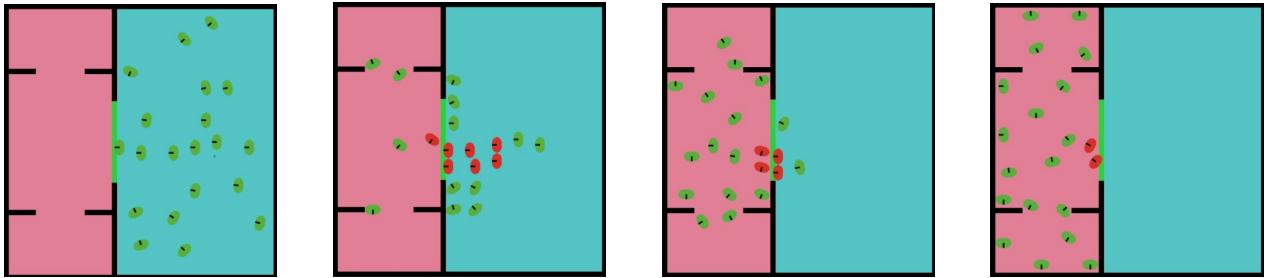


FIGURE 5 – Déroulement de la simulation pour 20 personnes polies et une patience de 20 ticks qui veulent monter, aux ticks 1, 32, 36 et 81.

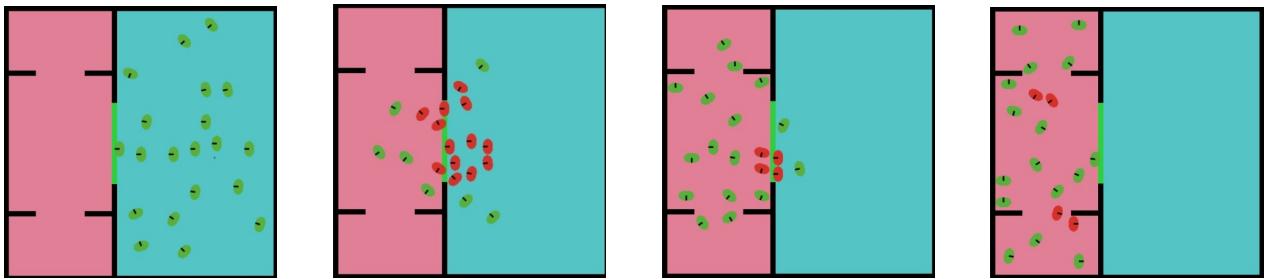


FIGURE 6 – Déroulement de la simulation pour 20 personnes impolies qui veulent monter, aux ticks 1, 20, 36 et 52.

Lors de ces simulations, on peut afficher le nombre de tortues qui sont montées dans le train selon le nombre de ticks, le nombre de tortues polies selon le nombre de ticks ainsi que le rayon libre moyen autour des tortues selon les ticks également.

Le rayon moyen libre autour des tortues représente leur confort. Il est au minimum de 3 car une tortue mesure 3 patchs. Plus il est élevé, plus les tortues ont du confort.

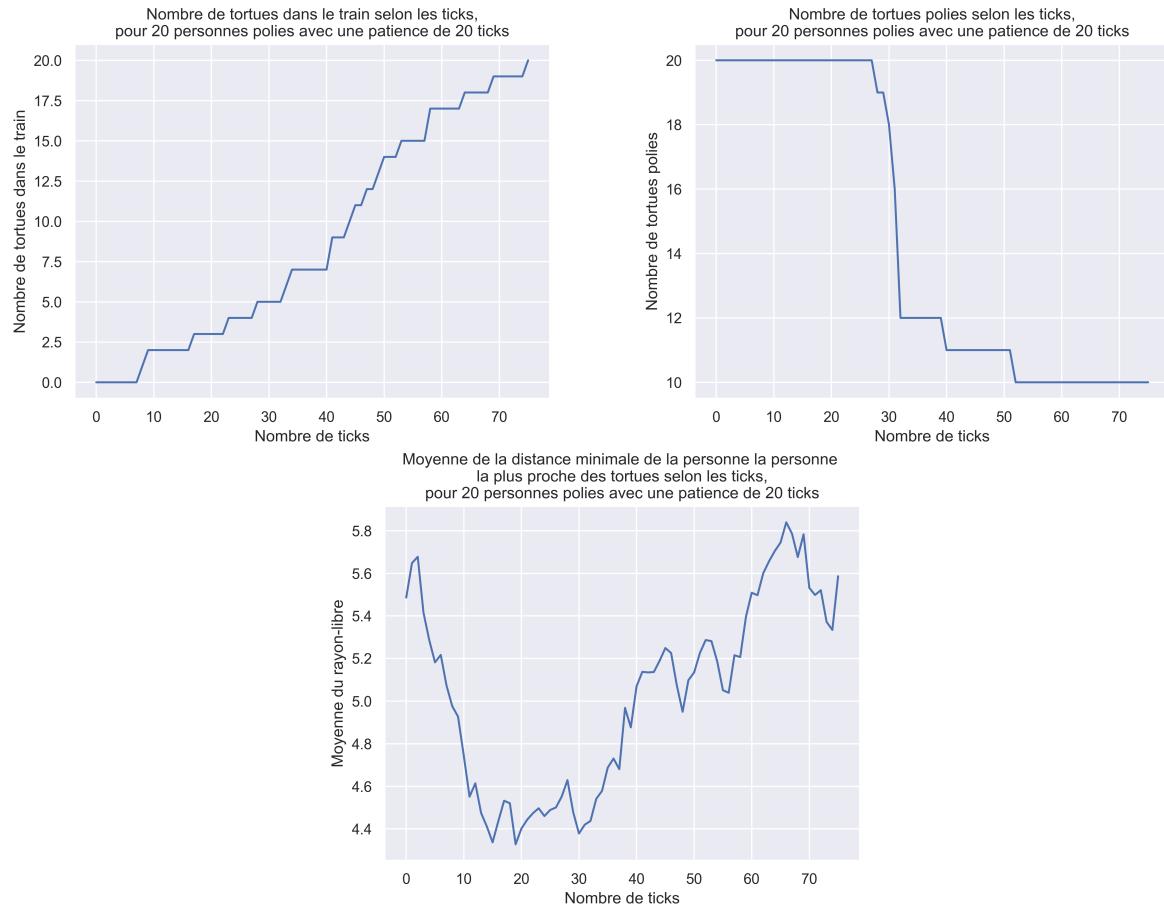


FIGURE 7 – Différentes statistiques pour une simulation avec 20 personnes polies et une patience de 20 ticks.

On peut analyser les images précédentes. Dans un premiers temps, jusqu'au tick 30, quelques personnes montent. Les autres sont arrêtées derrière, elles sont polies, il doit y avoir un blocage et elles n'osent pas avancer. Autour du 30e tick, le nombre de personnes polies chute, les tortues n'ont plus de patience, elles décident de forcer le passage, en collant les autres d'où la rupture de pente sur le nombre de personnes qui sont dans le train à partir du 40e tick, jusqu'à ce que tout le monde soit monté.

Le rayon-libre moyen est fort au début, quand les tortues sont éparpillées sur le quai. Il diminue (mais reste au dessus de 4 car les tortues sont polies) quand les tortues se rapprochent de la porte, puis il augmente de nouveau quand les tortues rentrent dans le train et peuvent s'écartier de nouveau.

Regardons une autre simulation avec plus de personnes pour voir ce que cela change.

Dans cette simulation, 80 personnes polies avec une patience de 20 ticks sont sur le quai et veulent

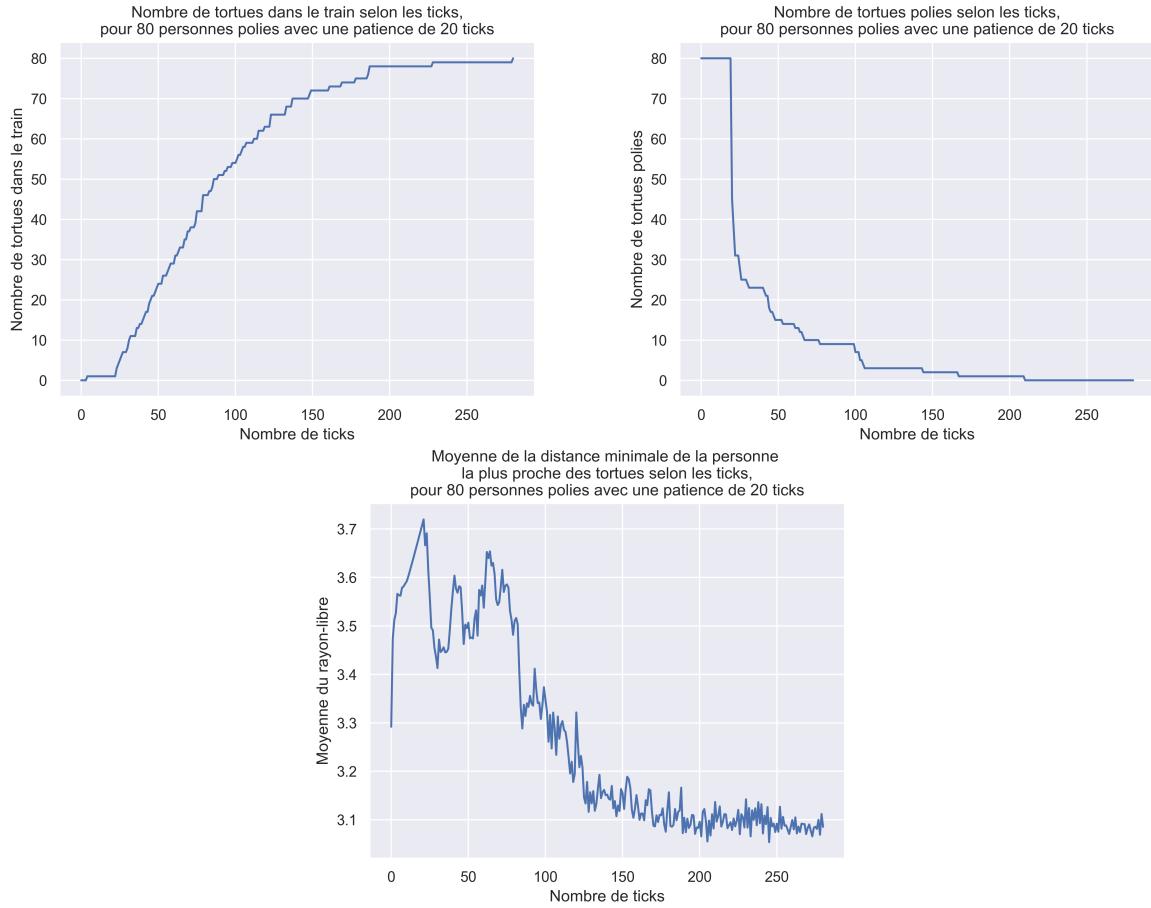


FIGURE 8 – Différentes statistiques pour une simulation avec 80 personnes polies et une patience de 20 ticks.

monter. Au début, à part une personne qui est juste en face de la porte et peut directement rentrer, rien ne se passe pendant 20 ticks, les tortues sont bloquées, elles n’osent pas avancer. Lorsque leur patience est finie, il y a une grosse chute de personnes polies et la situation se débloque. Le nombre de tortues dans le train augmente, d’abord linéairement puis le rythme ralentit. En effet, il est plus compliqué de trouver une place lorsqu’il y a beaucoup de monde dans le train. D’autant plus que les personnes dans le train ne sont pas coopératives, elles n’essayent pas de laisser de la place aux personnes qui montent, elles se dirigent simplement vers une place où elles auraient plus de place, en essayant de s’écartier des autres tortues. Ainsi, la dernière tortue qui monte prend beaucoup de temps, et on se retrouve avec une fonction convexe.

Au début de la simulation, la place disponible est déjà faible, avec un rayon-libre moyen de 3.3 patchs, puis lorsque les tortues montent dans le train, avec une surface moins grande que celle du quai, ce rayon descend à 3.1 patchs, les tortues sont donc très serrées, ce rayon valant au minimum 3 patchs.

Les KPI qu’on peut ressortir de ces simulations sont :

- le nombre de ticks avant que tout le monde soit monté
- le rayon moyen libre autour des tortues

On réalise des simulations en faisant varier le nombre de personnes sur le quai, en changeant la proportion de personnes polies au début de la simulation et en changeant la patience des personnes polies.

À cause de limite de capacités numériques, voici les résultats que l'on obtient avec 10 runs par cas traité. L'intervalle de confiance à 95% est calculé à partir d'une loi de Student.

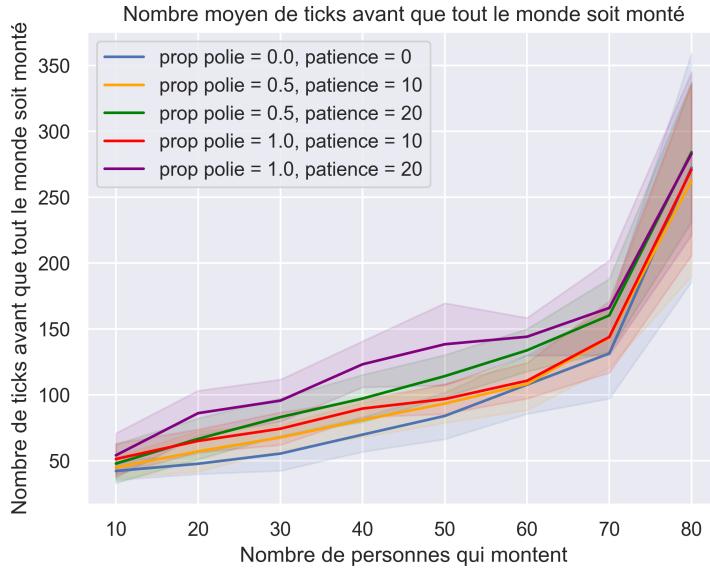


FIGURE 9 – Nombre moyen de ticks avant que toutes les personnes soient montées

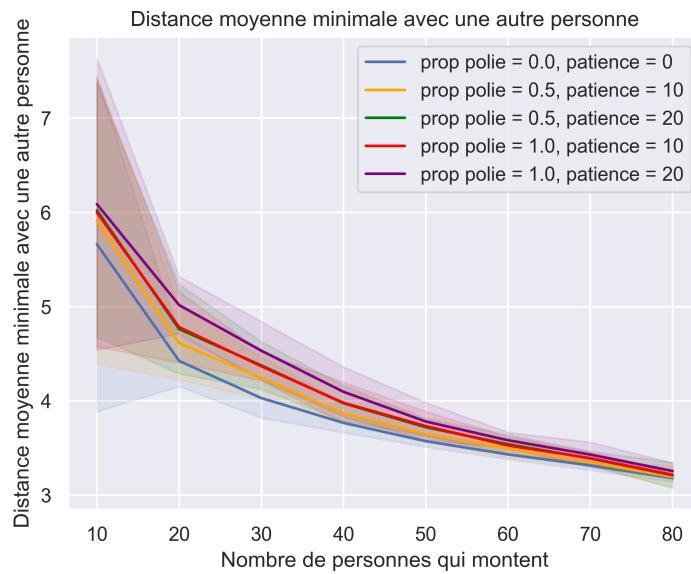


FIGURE 10 – Rayon moyen libre autour des tortues

Lorsque le train est rempli, il contient un peu plus de 80 personnes. Cependant, les dernières personnes ont beaucoup de mal à rentrer, ce qui explique l'explosion de temps quand on arrive à cette valeur.

On peut voir dans les graphiques ci-dessus que moins il y a de personnes polies, plus la montée se déroule rapidement. C'est principalement dû à cause de blocage lorsque les personnes sont polies, et cela augmente le nombre de ticks de leur patience, le temps qu'elles deviennent impolies et acceptent de se coller à d'autres tortues pour avancer.

Lorsqu'il y a peu de personnes, le rayon moyen libre est plus grand dans le cas où les tortues sont polies. Bien que la montée soit plus lente, on peut penser que si elle reste en dessous du seuil normal d'arrêt, cela ne dérange pas les tortues, puisqu'elles arrivent à monter dans le train, et cela en gardant un plus grand confort.

Pour les temps de montée, nous avons parlé uniquement de ticks, il faudrait parler en secondes pour pouvoir appliquer ce modèle dans la vie réelle.

Dans la littérature, notamment dans la thèse de F.Elleuch [1], la valeur de  $1.3m.s^{-1}$  apparaît, comme vitesse de marche moyenne. Nous ne savons pas si cette vitesse est adaptée sur une interface quai/train, mais c'est la vitesse que nous allons choisir.

Une personne peut avancer d'un patch par tick, donc de  $0.13m.tick^{-1}$  d'où 1 tick vaut  $\frac{0.13}{1.3} = 0.1s$ . Cela nous donne un temps de moins de 30s pour faire monter 80 personnes dans une voiture vide, d'après la figure 9. Sans données, nous ne pouvons pas estimer la cohérence de ces résultats. Toutefois on peut penser que le fait de faire 10 "mini pas" en 1s au lieu de faire un seul grand pas en 1s limite le phénomène d'accordéon qui est observé dans la littérature.

On peut observer une rupture de pente de pente à partir de 70 personnes qui veulent monter pour une capacité maximale de 80 personnes, donc environ 87.5% de la capacité maximale totale.

Si l'on voulait affiner le modèle, il faudrait prendre en compte qu'en réalité, les gens n'avancent pas de 0.13m toutes les 0.1s, leurs déplacements sont moins fluide.

De plus, les personnes se déplacent à des vitesses différentes, c'est un critère que nous voulions prendre en compte mais n'avons pas réussi à implémenter sur Netlogo, car nous nous sommes préoccupés à avoir des scénarios visuellement cohérents avant de prendre en compte cela.

D'autres phénomènes qu'on aurait aimé inclure dans les scénarios sont les effets de groupe pour étudier leur influence. En effet, les personnes ne prennent pas forcément le train seules, certaines personnes peuvent se suivre, des familles ou amis.

De plus, certaines personnes avec des bagages prennent plus de place que des passagers normaux, nous aurions aimé étudier leur influence également.

Ce premier scénario prend en compte l'espace entre les personnes, il faudrait l'adapter dans le cas des scénarios de descente avec obstacles et dans le cas de montées/descentes, que nous présentons ci-dessous.

### 3.4 Descente

Dans cette section, nous détaillerons le second scénario mis en place, qui correspond à la descente du train. On considère qu'il n'y a des tortues qu'à l'intérieur du train et on observe leur descente sur le quai. L'objectif est d'analyser le comportement des agents, collectif et individuel, et de voir leurs réactions à des perturbations.

Dans cette situation, on n'a pas besoin de modéliser la compression. En effet, on va considérer que tous les agents ont un comportement coopératif, puisqu'ils ont tous le même objectif : sortir du train et

atteindre la sortie. Ainsi, on ne parlera pas ici du coefficient de politesse et de patience, et on considérera donc qu'ils valent chacun 0. De plus, on va considérer que les agents sont prêts à sacrifier leur confort au profit de l'objectif de sortie. On ne parlera donc pas de compressibilité puisque la zone de confort n'est plus pertinente dans ce cas-là. Les tortues s'agglutinent devant la porte dans l'attente de sortir. La modélisation de la situation initiale, illustrée en figure 11, suppose que les tortues prêtes à sortir sont réparties autour des portes, ce qui semble réaliste.

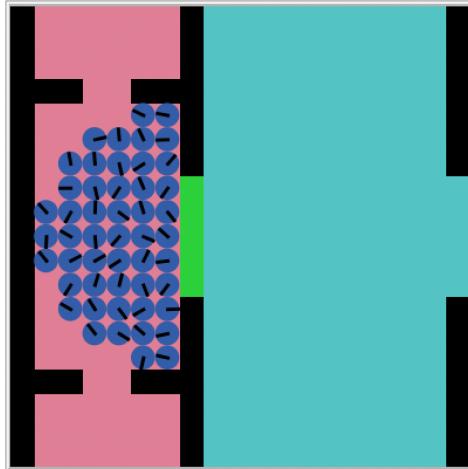


FIGURE 11 – Répartition initiale des tortues pour 50 passagers

On supposera dans cette modélisation qu'il n'y a qu'une seule sortie, et qu'elle se situe face à la porte du train. Par soucis de simplicité, puisque l'on a plus besoin de modéliser la compressibilité, on modélisera les tortues par des cercles de taille *1 patch*. Nous nous placerons dans le même cadre que pour le scénario de montée, en adaptant les distances, soit une longueur de train  $L = 18$  patchs et une largeur de  $l = 7$  patchs. La porte mesure 2m, donc  $w = 5$  patchs de largeur. On centre la porte, et on ajoute des murs pour avoir les 3 zones dans le train, devant la porte, à gauche et à droite. On dira que la sortie fait la même taille que la porte du train.

Supposons d'abord que le quai est vide, et que le train n'est composé que d'individus voulant descendre. On pourra observer en figure 12 le déroulement d'une descente pour **50 personnes** initialement présentes.

On observe une détente des agents à partir du début du groupe jusqu'à ce que tout le monde soit sorti. Ici, on a laissé les tortues entassées au niveau de la sortie à la fin de la simulation. Cette dernière se termine quand la dernière tortue a quitté le train.

On peut alors tracer le temps de sortie de tous les passagers en nombre de ticks en fonction du nombre qu'ils sont (cf figure 13).

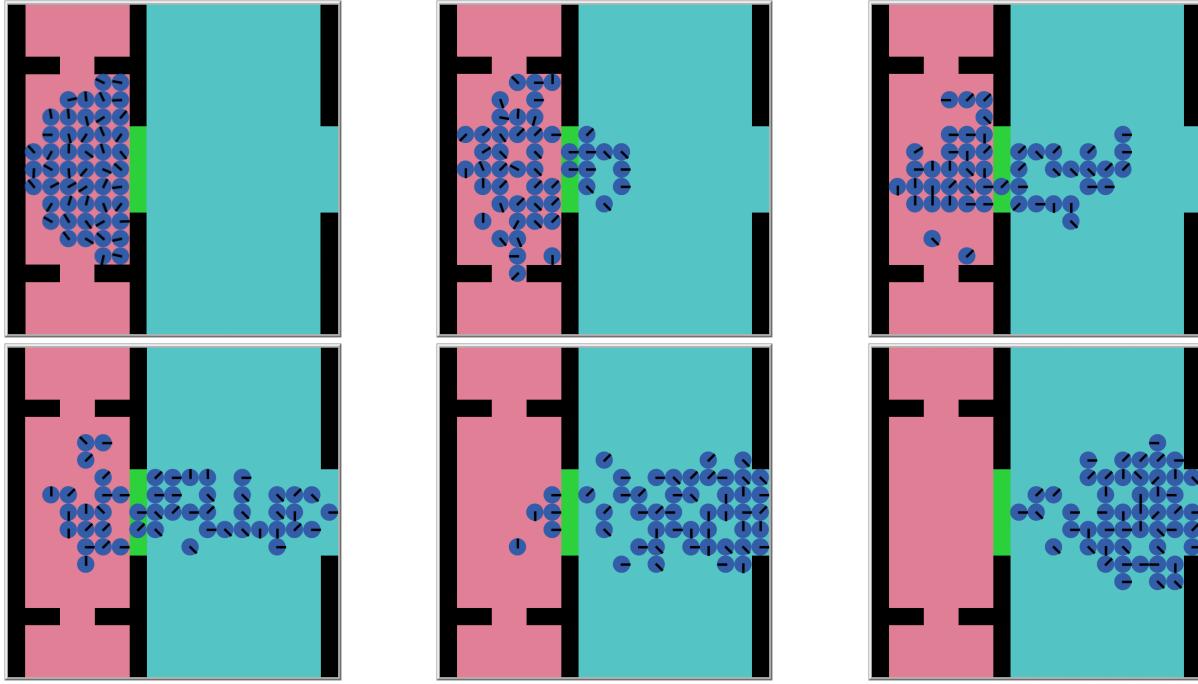


FIGURE 12 – Déroulé d'une simulation de descente simple pour 50 agents

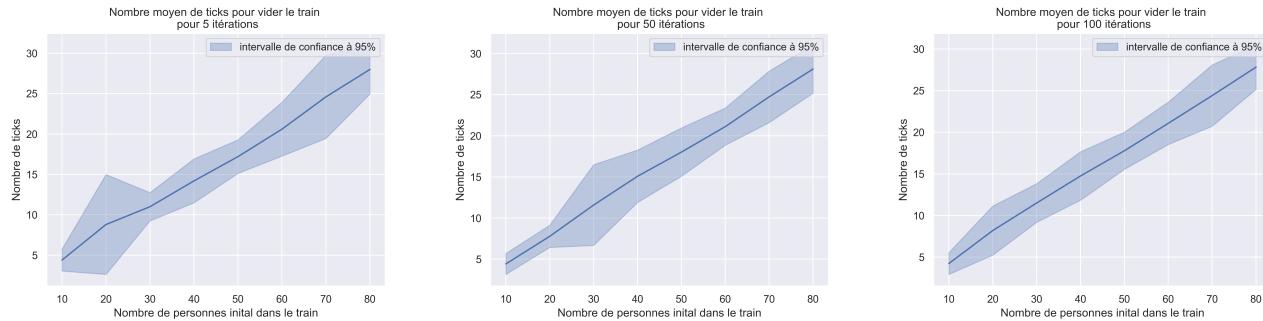


FIGURE 13 – Durée de la descente en fonction du nombre de personnes à bord et intervalles de confiance pour différents nombres de runs

La courbe tracée semble linéaire dans le cas où il n'y a pas d'obstacles, ce qui est plutôt cohérent. L'intervalle de confiance est plus précis avec plus d'itérations. Cependant, même avec un très grand nombre d'itérations, on arrive pas à faire beaucoup mieux que celui obtenus pour 100.

Pour se rapprocher plus de la réalité, on suppose dans un second temps qu'une partie des agents à bord du train y reste, alors que l'autre essaie de sortir. Dans ce cas, on repartit aléatoirement des agents dans le train et on les supposera **statiques**. Ensuite, on place les agents souhaitant sortir du train autour de la porte comme précédemment, mais en contournant les tortues statiques. Ces dernières constitueront des obstacles pour la sorties. On pourra voir par exemple de cas où 50 tortues sont présentent et 30 seulement veulent sortir.

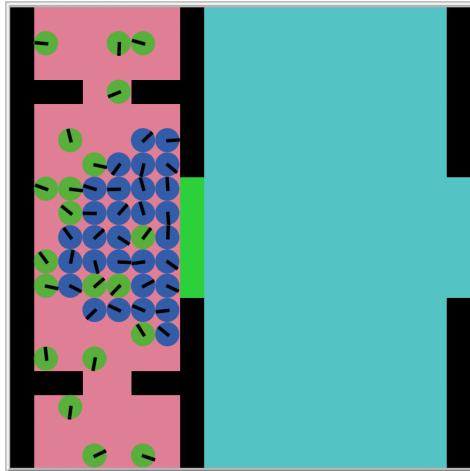


FIGURE 14 – Répartition initiale des tortues pour 50 passagers, dont 20 statiques en vert

Dans ce modèle, pour un nombre de personnes descendant fixé, et un nombre de personnes obstacles qui varie, la différence de ticks n'est pas frappantes (cf figure 15).

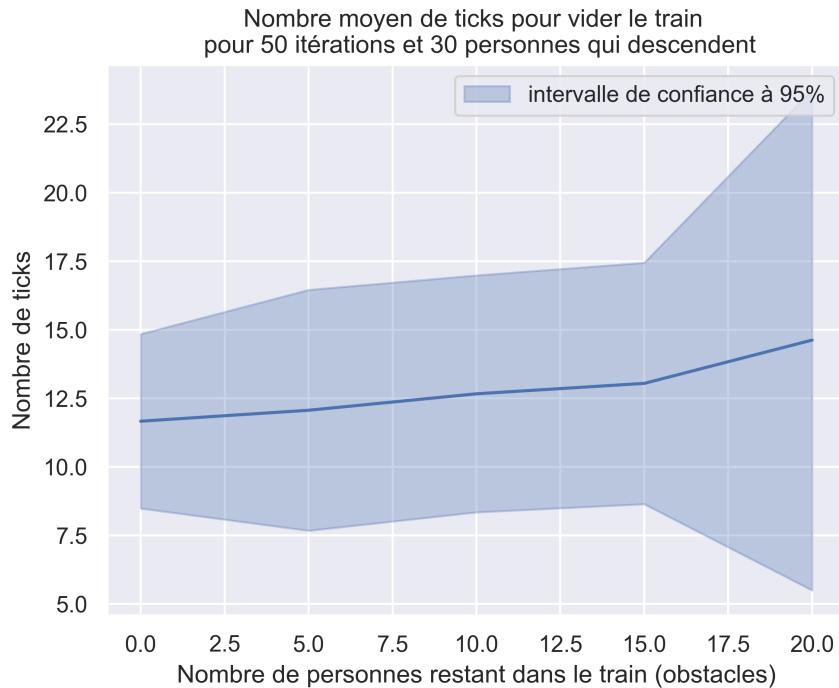


FIGURE 15 – Temps de descente en fonction du nombre de tortues immobiles et intervalle de confiance

Ce modèle est correct lorsque la proportion de gens statiques est relativement faible mais a très vite des limites. En effet, les intervalles de confiance explosent au delà de 20 tortues statiques ici et cela s'explique simplement. Le caractère immobile des tortues bloque certains patchs, qui ne sont plus accessibles et ainsi certains agents se retrouvent enclavés, faisant tendre la temps de simulation vers l'infini (en réalité ce sera le nombre de ticks maximum fixé), cf figure 16.

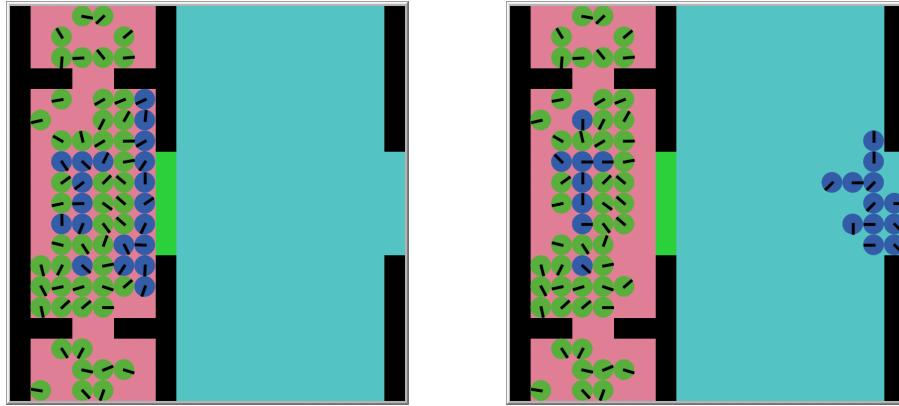


FIGURE 16 – Deux tortues bleues sont bloquées entre les tortues immobiles vertes

Dans l'idéal, il faudrait faire un modèle dynamique dans lequel les tortues s'écartent lorqu'elles bloquent une tortue voulant sortir (principe de priorité).

Nous étudierons alors un troisième cas dans le cadre du scénario de descente, celui des obstacles sur le quai. De la même manière, nous supposerons la répartition des tortues uniforme sur le quai. Cela peut être éloigné de la réalité car les individus sont souvent agglutinés autour des portes mais cette situation sera étudiée plus précisément dans le scénario d'échange. Ici, on supposera que les individus à quai ne veulent pas monter dans le train et qu'ils ne bougent alors pas.

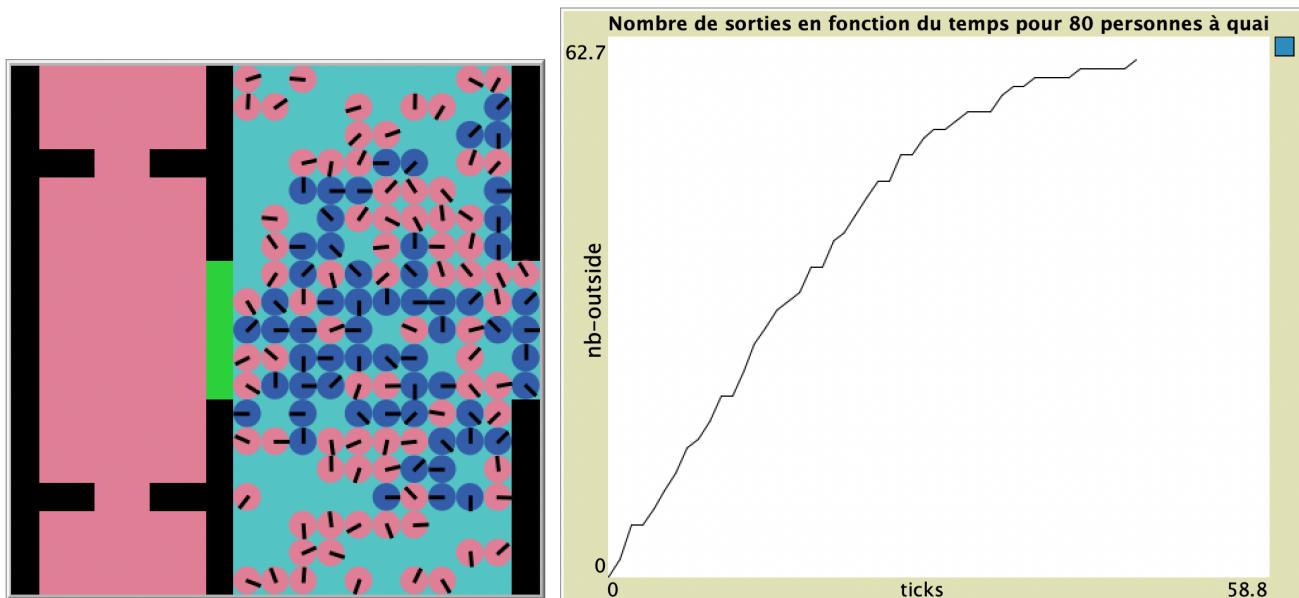


FIGURE 17 – Simulation pour 60 personnes descendant et 80 personnes à quai

Dans 17, on voit bien que la sortie s'effectue linéairement au début et que, à la fin, les tortues prennent plus de temps à sortir, d'où la courbe qui s'aplatis.

Ce résultat représentait une simulation, on en réalise maintenant plusieurs pour avoir des intervalles de confiance.

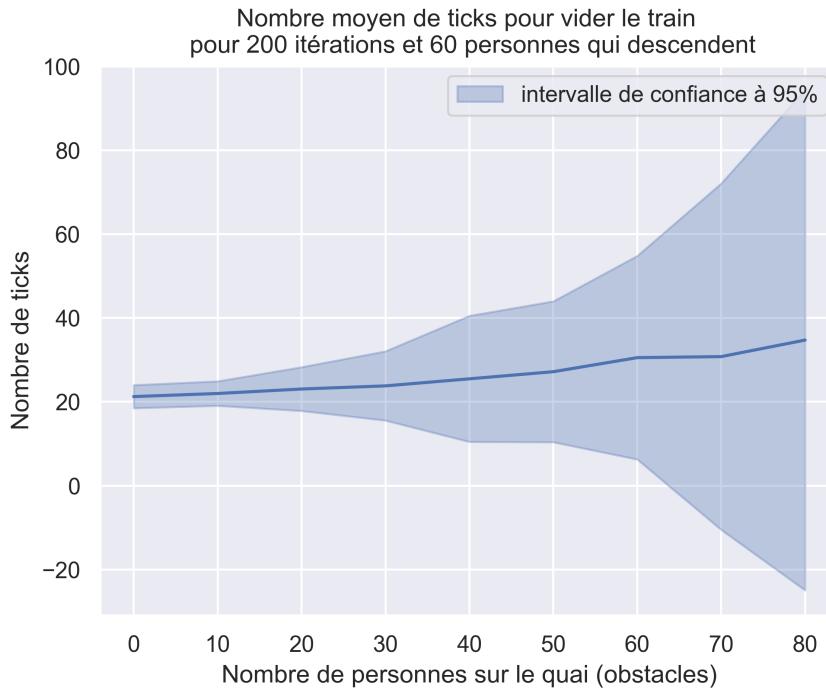


FIGURE 18 – Temps pour vider un train en fonction de l'affluence sur le quai et intervalle de confiance

Le courbe 18 obtenue est plutôt constante lorsque le nombre de personne sur le quai n'est pas trop important, avec un intervalle de confiance qui devient très grand dans le cas extrême où le nombre d'agent est maximal. Cela s'explique de la même manière que précédemment quand des agents restent bloqués. On observe un phénomène d'embouteillage lorsque le nombre total d'agents dans la simulation devient trop grand. En effet, on n'évacue pas ici les passagers à la sortie et on a donc un entassement.

Ces simulations peuvent nous amener à conclure que la descente ne pose pas de problème de manière générale. La seule limite est la capacité du quai ou du train et il faudrait améliorer le modèle de staticité pour coller plus à la réalité. De plus, on pourrait penser à un nouveau paramètre à faire varier : la largeur de la porte, puisque les personnes attendant à l'extérieur réduisent dans la réalité l'espace apparent où les personnes peuvent passer.

### 3.5 Échange

Dans ce scénario nous avons essayé de modéliser les comportements sociaux qui régissent les échanges au niveau de l'interface quai-train. Nous étudions principalement l'impact de ces comportements sur les temps d'échange sur cette interface.

Chaque personne est représenté par un agent et fait la taille d'un patch. Ainsi, à chaque pas de temps la personne avance, quand cela est possible, d'un patch dans son voisinage de Moore. Le voisinage de Moore d'un patch est constitué de la cellule centrale et des huit cellules adjacentes à celle-ci (diagonales comprises). Or, il y a un seul espace pour tous les agents au voisinage, en découle alors un phénomène de concurrence. La personne ne peut se mettre sur un patch que s'il est le plus prioritaire.

On se concentre sur deux aspects qui déterminent cette priorité, selon les tendances sociales observées lors de ces échanges. D'abord, chaque personne est plus ou moins capable de forcer son passage, parfois au détriment des autres passagers. On représente cette tendance par un indice d'impolitesse, compris entre 0 et 1. L'indice d'impolitesse est égal 0 pour les personnes très coopératives, 1 pour les autres. Par ailleurs, il y a souvent des passagers qui influent sur le comportement des autres en imposant un certain respect, par exemple, les personnes âgées, les femmes enceintes, etc. On modélise cette influence par un indice de diffusion de respect compris entre 0 et 1. Cet indice est égal à 1 pour les personnes diffusant un maximum de respect.

On a donc assigné à chaque agent un indice d'impolitesse noté  $I$ , et de diffusion de respect noté  $R$ . Ces deux indices régissent la concurrence sur un patch. En effet, on classe tous les agents susceptibles d'occuper un patch selon  $R/2 + I$ . Cet critère est choisi à partir de nos observations et est le plus proche du comportement qu'on souhaite modéliser.

Dans notre modèle, comme dans la réalité, les personnes qui descendent du train sont prioritaires, ainsi leur indice d'impolitesse est initialisé à une valeur entre 1 et 0.8. Les personnes sur le quai ont des valeurs initiales entre 0 et 0.2.

On se place dans le cadre représenté sur la figure 19, pour rappel, les patchs roses sont la voiture, les patchs bleus le quai, en vert la porte, le jaune les places assises et en noir, les murs qui sont impossible à traverser. On fait ensuite apparaître un certain nombre de tortues qui souhaitent monter (en rouge), descendre (en bleu), ou qui restent à leur place (en vert dans le train et en rose sur le quai). Afin de suivre le comportement des personnes atypiques, on les donnent des couleurs légèrement différentes. Ainsi, les passagers qui ne suivent pas un comportement « normal », sont en orange sur le quai et en bleu clair dans le train. De même, les personnes prioritaires sont en marron sur le quai et en violet dans le train.

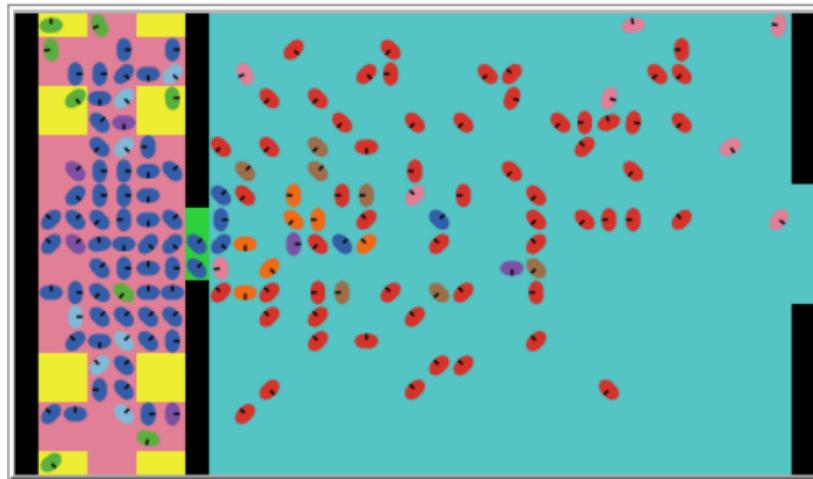


FIGURE 19 – Cadre du scénario montée et descente

Définissons maintenant le comportement des tortues. Chaque tortue qui se déplace vise en premier lieu la porte, ensuite elles visent la sortie s'elles souhaitent sortir ou une place assise si elles souhaitent monter dans le train. En cas d'absence de place assise dans le dernier cas, elles essayeront d'avancer vers

le fond du train afin de libérer le passage dans la mesure du possible. Une fois son objectif défini, elle essayera d'avancer d'un patch dans son voisinage de Moore si elle peut. Elle regarde alors les patchs de son voisinage, elle décidera de se mettre sur le patch libre qui est le plus proche de son objectif. Cependant, comme on l'a déjà indiqué, cette tortue ne peut se mettre sur ce patch que si elle est la plus prioritaire. Ainsi, on classe toutes les tortues susceptibles d'occuper ce patch et on vérifie si la tortue en question est en effet prioritaire, seulement dans ce cas elle pourrait avancer.

Cependant, ce comportement impliquera l'apparition de tortues qui restent toujours bloquées. Donc, au bout d'un certain nombre de ticks sans aucun mouvement, elle devient de plus en plus impatiente et elle forcera son passage d'un patch à l'autre sans vérifier le critère de priorité. Cette condition permettra d'éviter les phénomènes de blocage autour de la porte. En effet, sans obstacles autour de la porte, notre simulation garantit que chaque tortue finit par atteindre son objectif.

On peut voir sur la figure la réalisation d'une simulation avec 30 personnes qui monte, 30 personnes qui descendant avec 10% de personnes ayant un comportement atypiques, 10% de personnes qui diffusent un niveau élevé de respect, et 10% de personnes qui ne souhaitent pas se déplacer.

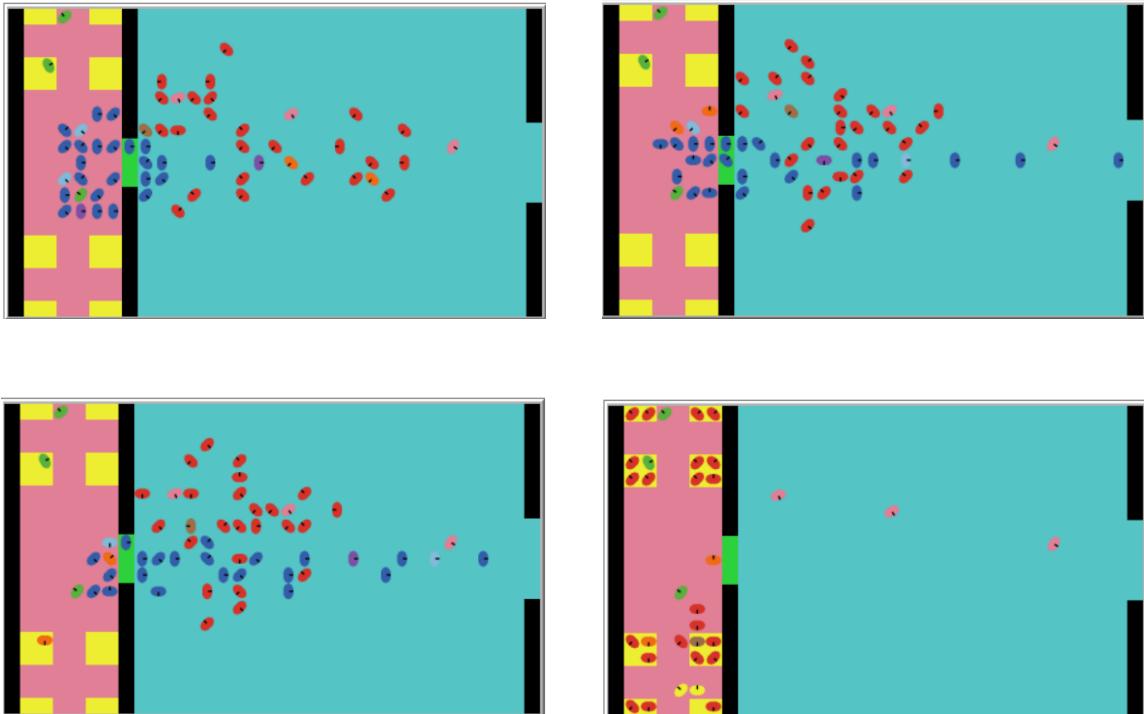


FIGURE 20 – Déroulement de la simulation pour 60 personnes, aux ticks 9, 41, 33 et 136 .

On remarque que les personnes souhaitent monter attendent que tout le monde sort avant de commencer la montée. Comme attendu, les descendants prioritaires descendent en avance (tortues violettes), les plus polis sont les derniers (tortues bleu clair). Pour la montée, les agents impolis montent avant la descente de tout le monde (en orange), alors que les autres attendent leur tour.

Dans le reste de nos simulations on se limite à un 2000 ticks afin d'obtenir des résultats en temps acceptable. On s'intéresse à la durée de l'échange, dès l'ouverture de la porte jusqu'à ce que tous les passagers atteignent leur objectif : descendre sur le quai ou monter dans le train.

On fait plusieurs simulations avec des proportions de comportements différents. Notre scénario de base comprend 10% des personnes ne souhaitant pas changer d'état, ils représentent donc des obstacles. On fait varier la proportion des personnes qui ne respectent pas le comportement prioritaire décrit précédemment et on observe l'évolution de la durée d'échange. On se limite dans notre étude dans le cas où le nombre de personnes qui descendant est égal à celui des personnes qui montent. On parle de ce nombre (pour chaque catégorie) dans ce qui suit.

Il faut noter que la durée de l'échange dépend aussi de la répartition initiale des personnes sur le quai et dans le train, plus particulièrement des personnes qui ne bougent pas et que l'on considère comme des obstacles. En effet, quand il y peu d'agents présents, il est facile de les détourner, c'est de plus en plus dure en augmentant le nombre d'agents étudiés.

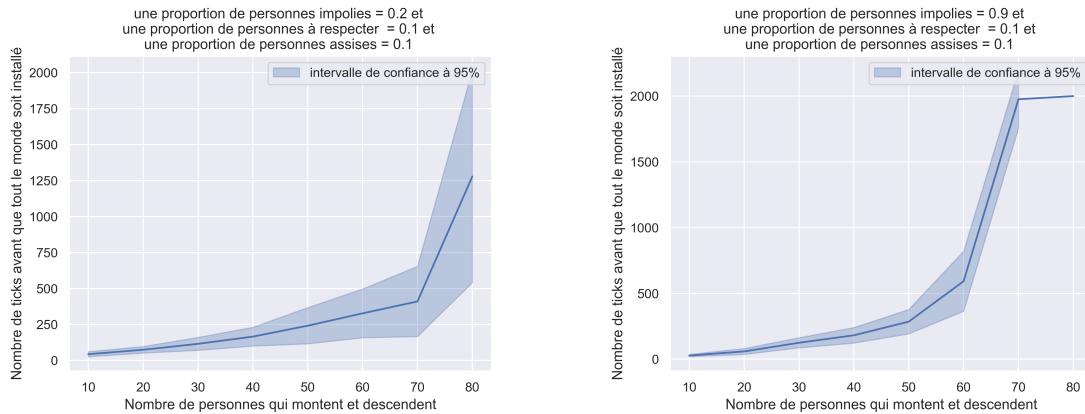


FIGURE 21 – Temps d'échange pour une proportion de personnes ayant un comportement atypique 20% et 90%

On peut remarquer l'existence de deux régimes. En premier lieu, les déplacements se font linéairement, le nombre moyen de ticks est proportionnel au nombre de personnes qui montent qui descendant. En deuxième lieu, on relève une rupture de pente. En effet, à partir d'un certain nombre d'agents, les échanges varie de façon exponentielle. Cette rupture est d'autant plus marquée quand on est dans le cas de scénario avec des personnes qui ne sont pas "polis" et ainsi ne respectent pas les priorités (figure 22) . Par exemple, dans le cas où le train est déjà plein, si les personnes qui attendent sur le quai essayent de monter et s'entassent sur la porte, alors que ceux qui veulent descendre ne sont pas sortis, on observe des situations de blocage importants.

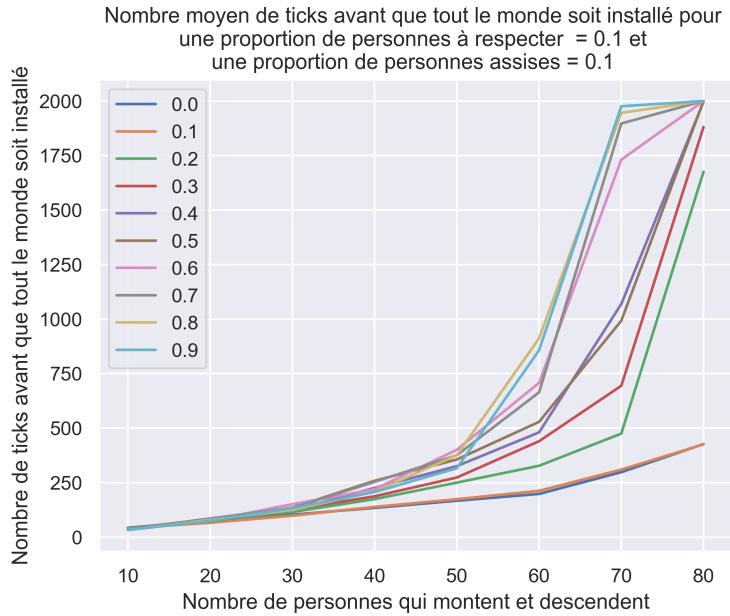


FIGURE 22 – Variation du temps de montée en fonction des proportions de personnes suivant un comportement atypique

S'il y a une minorité de personnes avec des comportements atypiques, l'échange se fait toujours en temps borné même dans le cas de 80 personnes (capacité totale du train). Par contre, ce même scénario conduit à des blocages importants dès que les personnes qui montent ou qui descendent dépassent 75% .

On a donc démontré que le comportement des passagers influe fortement sur le temps d'échange dans l'interface quai-train. Il serait intéressant d'étudier plus profondément les échanges au point de changement de régime. En outre, on pourrait étudier d'autres paramètres sociaux tels les comportements coopératifs entre les voyageurs qui peuvent eux aussi avoir un impact sur les retards observés lors des échanges devant les portes du train.

## 4 Recul sur les résultats et comparaison au cas réel

Nous avons à notre disposition pour ce projet

- notre modèle
- une base de données sur les temps d'arrêt et les flux de passagers dans une gare d'Île-de-France peu fréquentée

On peut estimer, après calculs sur moyenne sur la base de données, que le temps d'arrêt moyen est de l'ordre de 55 secondes. Néanmoins, sans détails supplémentaires il est difficile de connaître le temps d'échange en fonction du temps d'arrêt : un train peut rester à quai pour d'autres raisons que l'attente de la fin de l'échange des voyageurs. Nous pouvons donc uniquement savoir que le temps d'échange moyen dans cette gare est inférieur à 55 secondes.

Malheureusement les données ne sont pas assez riches, sur tout le mois il y a eu au maximum 362 personnes en simultané dans le train pour une capacité de 1844 personnes, soit à peine 20%. Il est difficile d'estimer l'impact de la charge du train sur les temps d'échange lorsque le train est aussi vide que celui ci. On peut voir sur la figure 23 que la gare en question, comptabilise beaucoup de montants le matin, de descendants le soir. On voit bien les deux heures de pointe.

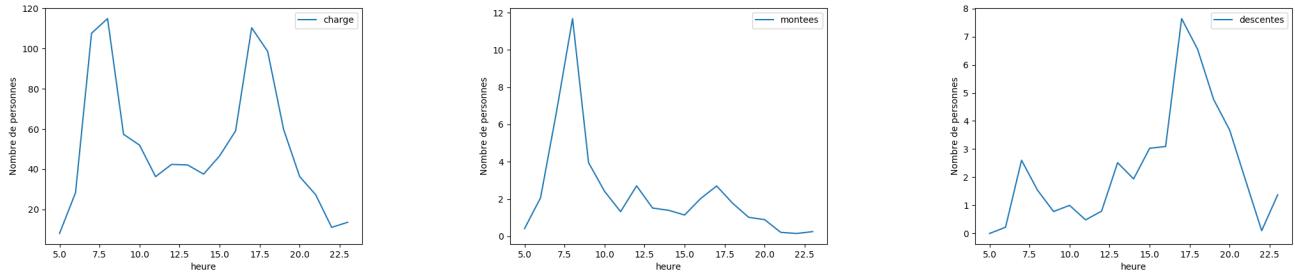


FIGURE 23 – Nombre de personnes à bord, montant, ou descendant lors d'une journée en moyenne sur la gare

## 5 Conclusion

Nous avons réussi à identifier un changement de régime sur la fonction du temps d'échange en fonction du nombre de personnes montantes et descendantes, et l'impact de la charge à bord.

Ainsi l'objectif qualitatif du projet a été accompli, même si beaucoup d'hypothèses simplificatrices ont été réalisées mais justifiées. Néanmoins, l'aspect quantitatif, quant à lui, a du être revu, faute de données et de littérature à ce sujet, il est assez difficile de réaliser un lien précis et robuste entre notre modèle et la réalité. La mesure du temps utilisée ici (les ticks) peut être reliée au temps réel grâce à un calcul d'ordre de grandeur et en se basant sur la vitesse moyenne de marche d'un homme, néanmoins on ne peut ni infirmer ni confirmer notre résultat pour l'instant.

Nous avons développé ci-dessous plusieurs axes possibles d'amélioration :

- un déplacement plus intelligent des personnes n'ayant pas d'objectif de montée ou de descente (dans nos simulations, elles restent immobiles)
- Modéliser le fait que les voyageurs ont des tailles différentes (avec objets : poussettes, sac à dos, valises...)
- Modéliser la présence de sous-groupe, de l'ordre familial ou amical, de personnes. Nous n'avons pas eu le temps de développer cette partie mais nous souhaitions représenter cela à la manière de nuées d'oiseaux *boids*(avec une personne devant représentant le leader)
- Au lieu de considérer qu'une personne avance d'un patch par tick, partir du principe qu'un tick vaut une seconde (donc un homme ne peut réaliser qu'au plus un pas par seconde, ce qui est la moyenne lors d'une marche). On pourrait ensuite faire varier la distance auquel les gens se déplacent, en moyenne  $1.3m.s^{-1}$  mais cette vitesse peut être plus moins grande selon les personnes (une personne jeune est plus rapide qu'une personne plus âgée). De plus, cette vitesse pourrait augmenter quand les personnes entendent le bip sonore des portes. Cela complexifie le code, mais est une voie d'amélioration pour plus de réalisme.
- Dans nos simulations, les paramètres de politesses/impolitesses sont fixés dès le début de la simulation sans prendre en considération les autres paramètres. Or on peut penser que lorsqu'il y a peu de personnes et donc que tout le monde va monter, les personnes se montrent polies mais lorsqu'il y a beaucoup de monde ou qu'il ne leur plus beaucoup de temps pour monter, elles deviennent impolies et auront plus tendance à pousser et forcer le passage.
- Prendre en compte la volonté des personnes de ne pas se coller dans les scénarios de descente et d'échange.

Malgré la simplicité apparente du simulateur multi-agents NetLogo, nous avons réussi à modéliser des comportements sociaux complexes à partir de règles simples. Cependant, il peut être un intéressant de chercher un outil de simulation multi-agent plus performant et plus facile à manipuler afin d'approfondir cette étude.

Nous souhaitons remercier Mostepha Khouadjia et Eric Duceau pour leur encadrement tout au long du projet et pour leurs bons conseils.

## 6 Bibliographie

### Références

- [1] Fatma ELLEUCH. « Transférabilité d'une modélisation-simulation multi-agents : le comportement inter-gares des voyageurs de la SNCF lors des échanges quai-train ». Thèse de doct. Conservatoire national des arts et métiers - CNAM, 19 mars 2019. URL : <https://tel.archives-ouvertes.fr/tel-02137524> (visité le 16/05/2022).
- [2] Marc JAXA-ROZEN et Jan H. KWAKKEL. « PyNetLogo : Linking NetLogo with Python ». In : *Journal of Artificial Societies and Social Simulation* 21.2 (2018), p. 4. ISSN : 1460-7425. DOI : 10.18564/jasss.3668. URL : <http://jasss.soc.surrey.ac.uk/21/2/4.html>.
- [3] Bachar KABALAN. « Crowd dynamics : modeling pedestrian movement and associated generated forces ». Thèse de doct. 12 jan. 2016.
- [4] OLIVIER RAZEMOND. *RER : le palmarès des stations les plus fréquentées [en ligne]*. 2016. URL : [http://ccl.northwestern.edu/netlogo/](https://www.lemonde.fr/blog/transport/2015/04/08/rer-le-palmares-des-stations-les-plus-frequeentes/#:~:text=Parmi%20les%20gares%20plus,%2019%2C8%20millions) ..</a></li><li>[5] Sebastián SERIANI et Taku FUJIYAMA. « Modelling the distribution of passengers waiting to board the train at metro stations ». In : <i>Journal of Rail Transport Planning & Management</i> 11 (août 2019), p. 100141. DOI : 10.1016/j.jrtpm.2019.100141.</li><li>[6] Uri WILENSKY. <i>NetLogo</i>. Rapp. tech. Northwestern University, Evanston, IL : Center for Connected Learning et Computer-Based Modeling, 1999. URL : <a href=).