Modélisation probabiliste de grands systèmes Praxitèle

Programme BigNet — Version 2.00

Jean-Marc LASGOUTTES

INRIA — Projet MEVAL

Mars 1997

Table des matières

1 Introduction

Ce rapport décrit des travaux effectués à l'INRIA au sein du projet MEVAL concernant la modélisation mathématique de réseaux de transport de type PRAXITÈLE. L'approche *probabiliste* utilisée permet de tenir compte des variations autour des valeurs moyennes que connaît tout processus réel et de remplacer les paramètres inconnus du modèle, qui sont nécessairement nombreux, par des valeurs aléatoires.

Contrairement à la simulation grace à laquelle on peut, au prix d'un temps de calcul important, calculer des mesures de performance de systèmes assez complexes, nous nous sommes intéressés ici à une classe de systèmes pour lesquels des résultats explicites peuvent être obtenus. Cette démarche, complémentaire à celle de la simulation, fournit des résultats se prêtant mieux aux calculs de dimensionnement.

Une autre particularité de ce travail est que nous nous sommes intéressés à la modélisation de *grands* réseaux. Le modèle que nous obtenons est surtout valide quand le nombre de stations Praxitèle est élevé (en pratique, une dizaine de stations est suffisante). Nos résultats peuvent donc être utilisés là où d'autres méthodes deviennent trop coûteuses en temps de calcul.

Nous avons obtenus des résultats permettant de calculer de manière simple les principales mesures de performance intéressantes. Ces résultats ont été implémentés dans le programme BigNet, qui permet de tracer des courbes montrant l'influence des différents paramètres sur le système.

Ce rapport est divisé en trois parties : dans la section 2, nous présentons notre modèle et discutons quels sont ses points forts et ses points faibles du point de vue de la modélisation de Praxitèle. Les principaux résultats obtenus sont présentés dans la section 3 ; on montre en particulier l'application de nos méthodes aux données modélisant Paris fournies par l'INRETS. Enfin, la section 4 est le manuel d'utilisation du programme BigNet.

2 Modélisation probabiliste d'un réseau Praxitèle

Le modèle présenté ici est en fait la spécialisation à Praxitèle d'un modèle général que nous avons étudié avec Guy Fayolle[1]. Le cadre est celui des réseaux fermés à forme produit, pour lesquels de nombreux résultats sont connus. Pour une description rigoureuse du modèle étudié et des résultats, nous encourageons le lecteur à se reporter à l'article cité ci-dessus, dans lequel l'application à Praxitèle est présentée.

On considère un réseau formé d'un nombre n de stations Praxitèle. Chacune de ces stations possède un parking de taille « infinie »[2] et est reliée aux autres par un réseau de routes. Dans ce réseau circulent un nombre fixé m de Praxicars qui peuvent être pris par des clients pour aller d'une station à une autre. On considère que cela peut se faire sans passer par une autre station et on néglige les interactions entre Praxicars : ils ne peuvent pas créer d'embouteillages sur les routes.

Le fonctionnement de ce système est le suivant : à chaque station arrivent des clients désirant prendre un véhicule. S'ils en trouvent un en attente, ils le prennent jusqu'à leur station de destination où il le laissent avant de quitter le système. Par contre, si la station est vide, ils n'attendent pas et repartent tout de suite.

1. Fayolle, G., et Lasgouttes, J.-M., « Asymptotics and scalings for large closed product-form networks via the Central Limit Theorem », *Markov Processes and Related Fields* 2(2), 1996, pp. 317–348

2. C'est-à-dire assez grand pour accueillir tous les Praxicars du réseau

Pour pouvoir donner un sens à ce modèle, il faut bien sûr énoncer les hypothèses mathématiques faites sur les processus en présence :

- les clients arrivent à chaque station selon un processus de Poisson[3] *stationnaire dans le temps*. Ce processus est défini par un taux d'arrivée (mesuré en nombre moyens d'arrivées par unité de temps) qui reste le même toute la journée ;
- la destination des clients est choisie aléatoirement à l'aide d'une table (matrice de transition) qui donne la probabilité d'aller d'une station à une autre ;
- la durée de déplacement est une variable aléatoire de moyenne connue, uniforme durant la journée et dépendant uniquement des station de départ et d'arrivée.

De manière équivalente, les taux d'arrivée et la matrice de transition peuvent être remplacés par une matrice Origine/Destination (OD), comme souvent en modélisation de trafic.

Les résultats que nous donnons ici sont valables en *régime stationnaire*, c'est-à-dire, après que le système a fonctionné pendant une durée suffisante. Cette durée peut par exemple être la juxtaposition de plusieurs journées. Nous n'incorporons pas ici de haut-le-pied dans le modèle : au contraire, nous cherchons à analyser le réseau livré à lui-même, pour en déduire où un contrôle serait nécessaire.

Même si ce modèle donne une bonne idée du comportement d'un système de type PRAXITÈLE, il faut toujours être conscient des caractéristiques qui sont simplifiées:

- *les matrices origine/destination sont constantes au cours de la journée* : c'est bien sûr la difficulté majeure, liée au fait qu'il est difficile de traiter des modèles significatifs variant au cours du temps. Cela conduit en général à des prévisions trop optimistes. On peut par exemple corriger cela en ne considérant que les tranches horaires les plus significatives et en prenant la matrice origine/destination *moyenne* sur ces tranches horaires. BIGNET propose aussi une méthode approchée permettant de prendre en compte ces variations, sans pour autant avoir l'exactitude des résultats avec des données constantes.

3. En particulier, à chaque instant, la date d'arrivée du prochain client ne dépend pas de la date d'arrivée du précédent. Ces processus sont souvent utilisés en modélisation à cause de leurs «bonnes» propriétés mathématiques. Il sont un choix raisonnable quand on ne dispose pas d'informations autres que les valeurs moyennes des taux d'arrivée.

– *les clients n'attendent pas si aucun véhicule n'est disponible* : ce choix peut paraître discutable, mais n'est a priori pas plus mauvais que de décider qu'ils attendent indéfiniment. La vérité est probablement quelque part entre les deux, mais, en l'absence de données précises, il est difficile de prendre en compte cette attente.

– *les parkings sont de taille infinie* : cette approche permet de mesurer la probabilité qu'un parking contienne plus de v véhicules et donc de déterminer une taille raisonnable à lui donner.

3 Principaux résultats sur des grands réseaux de type Praxitèle

3.1 Généralités

Dans cette section nous présentons les principaux résultats obtenus sur notre modèle. Il est possible de calculer, connaissant les paramètres du système et le nombre de véhicules utilisés :

– la proportion de clients perdus, globalement ou station par station ;

– l'usage des véhicules, exprimé en voyage par véhicule et par unité de temps ;

– le nombre de véhicules présents à chaque station ;

– le nombre *critique* de véhicules (à partir duquel en rajouter n'a plus d'intérêt),

– la taille à donner aux parkings pour assurer une probabilité de saturation donnée.

En fait, bien d'autres mesures de performance sont accessible, puisque nous calculons la distribution stationnaire du processus formé du nombre de clients à chaque station. Ces résultats sont en principe valables quand la taille du réseau et le nombre de véhicules tendent conjointement vers l'infini. Toutefois, l'erreur relative entre nos estimés et les vraies valeurs est en général inversement proportionnelle au nombre de véhicules. On peut donc espérer une bonne précision avec quelques dizaines de véhicules.

Il s'avère que le comportement d'un grand réseau de véhicules PRAXITÈLE peut être comparé à un système ouvert comportant les mêmes stations fonctionnant *en parallèle*, indépendemment les unes des autres. Il suffit, pour déterminer ce système équivalent, d'ajuster le flux de véhicules qui y arrivent de façon à ce que leur nombre moyen soit égal au nombre de véhicules dans le vrai réseau PRAXITÈLE. Ensuite, toutes les mesures de performance usuelles peuvent être calculées sur le système équivalent, qui est très simple à analyser.

Une conséquence un peu surprenante de ces résultats est qu'il existe une station particulière (ou parfois plusieurs) qui joue un rôle de goulot d'étranglement : si on augmente le nombre de véhicules PRAXITÈLE dans le réseau alors, à partir d'un moment, les véhicules ajoutés vont aller s'agglutiner dans cette station. Ils ne pourront donc pas être utilisés dans les autres stations, qui souffriront d'un taux de perte de clients incompressible. Le nombre de véhicules nécessaires pour atteindre cet état est le nombre critique cité plus haut.

La détection de ce goulot d'étranglement est donc très importante dans les tâches de dimensionnement. Il peut être fait de manière simple à partir uniquement de la matrice origine/destination et est en particulier complètement indépendant des temps de parcours, ainsi que la valeur minimale atteignable pour les taux de perte.

3.2 Indicateur de déséquilibre spatial

Une caractéristique intéressante des résultats présentés ici est qu'ils permettent de calculer de manière naturelle un indicateur de déséquilibre spatial du réseau[4]. Même si cet indicateur n'est pas d'un calcul immédiat, il peut être interprété physiquement dans le cadre ne notre modèle.

Un peu de notations vont être nécessaires pour définir cet indicateur. On note M la matrice origine/destination d'ordre n du modèle, où $M_{i,j}$ est le nombre de trajets entre les stations i et j. De plus, On note O_i le nombre total de trajets issus de i, c'est-à-dire

$$O_i = \sum_{j=1}^{n} M_{i,j}.$$

4. en ce sens que cet indicateur apparaît naturellement dans nos calculs théoriques

On définit alors les nombres $\delta_1, \ldots, \delta_n$ par les équations

$$\delta_j = \sum_{i=1}^{n} \delta_i \frac{M_{i,j}}{O_i O_j}, \qquad \sum_{j=1}^{n} \delta_j = 1.$$

Notre *Indicateur de déséquilibre spatial* de M est alors donné par la formule

$$\Delta = \max_{1 \leq j \leq n} \delta_j.$$

Cet indicateur a plusieurs propriétés intéressantes :

1° on a toujours $\frac{1}{n} \leq \Delta \leq 1$;

2° $\Delta = 1$ si et seulement si toutes les colonnes de la matrice sauf une sont nulles (système complètement déséquilibré) ;

3° $\Delta = \frac{1}{n}$ si et seulement si le taux de satisfaction est le même à chaque station (système complètement équilibré);

4° le taux de satisfaction est donné pour n assez grand par

$$\tau_{\text{sat}} = \frac{1}{\Delta} \frac{\sum_{k=1}^{n} \delta_k O_k}{\sum_{k=1}^{n} O_k}.$$

De plus, le calcul de cet indicateur, même s'il nécessite la résolution d'un système d'équations, reste assez simple et notablement moins coûteux en temps de calcul qu'une simulation.

3.3 Un modèle de Paris en 7 stations

Le premier exemple traité est issu de données agrégées fournies par l'INRETS sur la demande potentielle dans 7 arrondissements parisiens. Ces données étant initialement découpées par tranche horaire, et il a fallu les agréger à nouveau pour nos besoins.

Elles sont présentées dans les tables 1 et 2. Nous les avons utilisées pour trouver, d'une part les stations jouant un rôle limitant et, d'autre part, une estimation du nombre de véhicules nécessaire dans une telle configuration. La figure 1 montre l'évolution du du taux de perte de clients en fonction du nombre de véhicules dans le réseau. Elle a été produite directement par BIGNET.

Station	1	2	3	4	5	6	7
1	187	328	0	354	38	527	222
2	240	408	214	193	49	212	65
3	94	0	314	163	195	0	0
4	459	280	85	258	159	306	0
5	393	132	264	52	352	364	189
6	86	135	143	469	0	537	305
7	277	296	0	171	111	358	231

TAB. 1 – *Nombre de trajets de station à station en une journée (20 heures)*

Station	1	2	3	4	5	6	7
1	4	4	8	8	12	7	8
2	4	2	5	9	15	10	11
3	8	5	6	6	12	16	19
4	8	9	6	4	7	11	14
5	12	15	12	7	5	7	15
6	7	10	16	11	7	4	9
7	8	11	19	14	15	9	4

TAB. 2 – *durée d'un trajet de station à station en minutes*

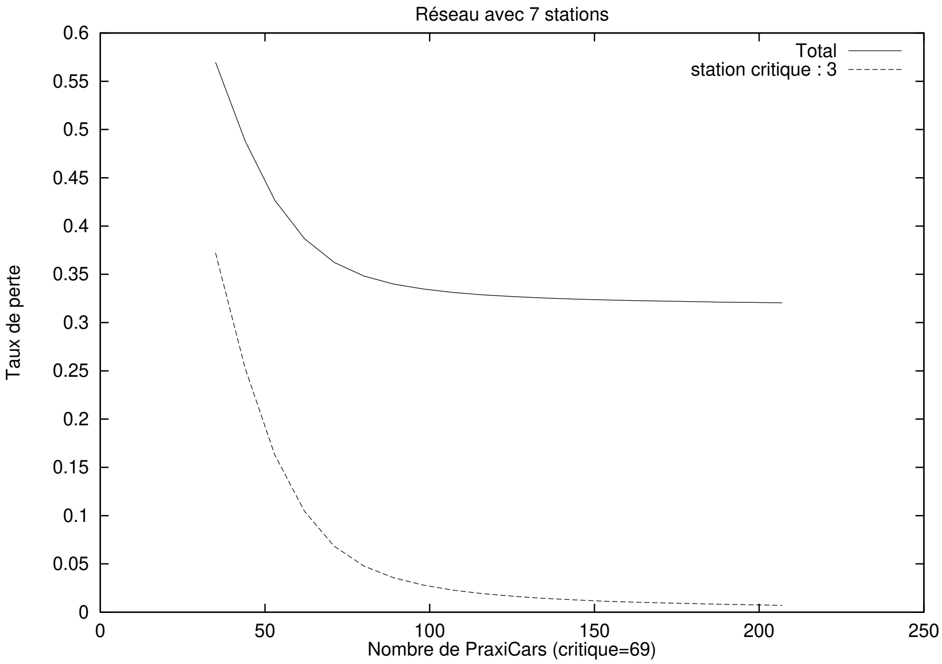

FIG. 1 – *Taux de perte en fonction du nombre de véhicules*

3.4 Un grand réseau en étoile

Le deuxième type de réseau présenté ici comporte une station centrale et n stations périphériques placées en cercle. Trois types de déplacements sont possibles :

- de la station centrale à une des stations périphériques, avec probabilité $1/n$;
- d'une station périphérique vers la station centrale, avec probabilité p $(0 < p < 1)$;
- d'une station périphérique à une de ses deux voisines, avec probabilité $(1-p)/2$.

L'intérêt de ce modèle est qu'il permet de comprendre l'influence d'une station centrale (par exemple une gare) avec un fort débit sur un ensemble de stations homogènes. De plus, il permet de fixer librement le nombre de stations et d'étudier le comportement de grands réseaux pour lesquels peu de données réalistes existent. Dans l'exemple de la figure 2, la taille du réseau est fixée a 100 stations, et le taux d'arrivée à la station centrale varie de 3 à 10 clients par unité de temps.

Cet exemple montre assez bien l'effet de saturation dans le réseau : quand le taux d'arrivée dans la station centrale est inférieur à 8.5, cette station est un goulot d'étranglement où les véhicules s'accumulent (taux de perte très faible). Au contraire, au delà de 8.5, ce sont les files périphériques (toutes identiques) qui se saturent. On observe donc que le taux de perte moyen présente un minimum en 8.5.

4 Le programme d'évaluation BigNet

4.1 Utilisation du programme

Le programme BigNet qui fonctionne sous MS-DOS ou MS-Windows, permet de calculer l'influence des paramètres du système (nombre de stations, nombre de véhicules, taux d'arrivée, etc.) sur sa performance (taux de perte, utilisation des véhicules, etc.). C'est un programme non interactif qui prend en entrée un fichier de configuration et donne en sortie un tableau de données et utilise éventuellement le programme GNUPlot pour afficher les courbes correspondantes.

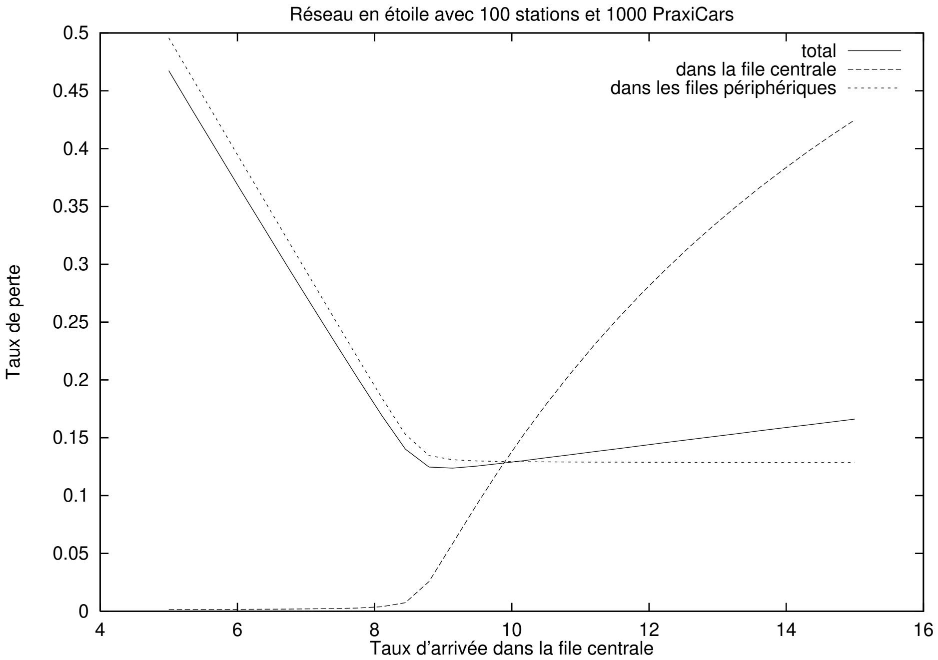

Fig. 2 – *Taux de perte en fonction du taux d'arrivée*

Toutes les courbes présentées dans la section précédente proviennent directement de BigNet. Il est toutefois possible de générer uniquement un tableau de données pour les utiliser autrement (avec un tableur ou un outil d'analyse de données par exemple).

L'utilisation de BigNet se fait en deux phases :

1. création d'un fichier de configuration décrivant le réseau ;
2. exécution de BigNet, soit à partir du gestionnaire de fichiers de Windows, soit par la commande `bignet.exe` ⟨*nom-du-fichier-de-configuration*⟩.

Si BigNet est appelé sans paramètre, un nom de fichier de configuration sera demandé au démarrage.

À chaque fois que BigNet est lancé depuis Windows et que des courbes sont demandées, il exécutera automatiquement `wgnuplot.exe` pour les afficher. Pour imprimer la courbe affichée ou modifier les paramètres (couleur, police de caractères, etc.), utiliser le menu qui apparaît en cliquant sur le bouton droit de la souris. Par contre BigNet n'exécute pas GNUPlot s'il est lancé sous MS-DOS ou dans une fenêtre DOS de Windows.

```
# Réseau en étoile
NetType       Star

#Nombre de stations Praxitèle
Stations      100
#Nombre de véhicules
PraxiCars     1000
#Taux d'arrivée aux files (centrale,périphériques)
ArrivalRate   5:15 1
#Durée de trajet (c->p,p->c,p->p)
Journey       10 10 2
#Routage (proba d'aller de périph. à centrale)
Routing       0.10

# Nombre d'itérations
Iterations    30
#Nom du fichier de sortie de données
OutputFile star.out
#Nom du fichier de commande GNUplot et titre du graphique
PlotFile   star.plt
Title "Réseau en étoile avec {Stations} stations et {PraxiCars} PraxiCars"
# Variable principale et titres sur les axes
PlotVar    ArrivalRate[1]
Title "Taux d'arrivée dans la file centrale|Taux de perte"
# Courbes à tracer avec leur titre
Plot       LossProba    Title "total"
Plot       LossProba[1] Title "dans la file centrale"
Plot       LossProba[2] Title "dans les files périphériques"
```

FIG. 3 – *Fichier de configuration pour un réseau en étoile*

Dans les sections qui suivent, nous décrivons la structure des fichiers de configuration de BIGNET.

4.2 Syntaxe du fichier de configuration

La définition d'un réseau se fait par une succession de variables auxquelles on associe une valeur. Par exemple, la définition du réseau en étoile de la section 3.4 se fait grâce au fichier de configuration présenté en figure 3.

Dans cet exemple, les lignes commençant par le caractère # sont des commentaires, ignorés par le programme. Les majuscules et minuscules sont équivalentes. La première partie du fichier décrit le réseau, alors que la deuxième décrit les données que l'on veut obtenir.

La syntaxe générale d'une définition de variable est

$\langle$*Variable*$\rangle$ $\langle$*valeur*$\rangle$ $\langle$*valeur*$\rangle$ ...

où $\langle$*Variable*$\rangle$ est le nom de la variable à définir et chacun des champs $\langle$*valeur*$\rangle$ peut être :

- un nombre entier,
- un nombre réel,
- un intervalle de nombres entiers, de la forme $\langle$*entier*$\rangle$:$\langle$*entier*$\rangle$,
- un intervalle de nombre réels, de la forme $\langle$*réel*$\rangle$:$\langle$*réel*$\rangle$,
- une chaîne de caractères, qui peut être éventuellement comprise entre guillemets (`"..."` ou `'...'`, les uns pouvant être contenus dans les autres comme dans `"aaa 'bbb' ccc"`).

Il est important de noter que toutes les variables qui acceptent une valeur de type intervalle acceptent aussi une valeur du type simple correspondant. Ces intervalles permettent de faire varier la valeur des différents paramètres du réseau à chaque itération.

Enfin, il est possible pour les variables de type réel ou intervalle de réel de spécifier un coefficient multiplicateur à l'aide de `*`$\langle$*réel*$\rangle$ ou `/`$\langle$*réel*$\rangle$:

$\langle$*Variable*$\rangle$`*`$\langle$*Coeff*$\rangle$ Indique que les valeurs suivantes doivent être *divisées* par $\langle$*Coeff*$\rangle$;

$\langle$*Variable*$\rangle$`/`$\langle$*Coeff*$\rangle$ Indique que les valeurs suivantes doivent être *multipliées* par $\langle$*Coeff*$\rangle$.

Les sauts de lignes, les virgules et les point virgules sont considérés comme des espaces, et plusieurs variables peuvent apparaître sur la même ligne. Il est de plus possible de lire des données dans un autre fichier avec la directive ReadFile $\langle$*nom-de-fichier*$\rangle$, qui peut même être utilisée à la place de la valeur d'une variable, ce qui permet de garder les données dans des fichiers texte simples (par exemple Routing ReadFile $\langle$*nom-de-fichier*$\rangle$ lira la matrice de routage dans le fichier $\langle$*nom-de-fichier*$\rangle$). Il n'est pas possible d'imbriquer les directives ReadFile.

4.3 Description du réseau

Le réseau Praxitèle à étudier est décrit à l'aide des variables suivantes :

NetType ⟨*nom*⟩ où ⟨*nom*⟩ est le type de réseau à étudier, à choisir parmi Custom (quelconque), Star (en étoile) et Symmetrical (complètement symétrique). La variable NetType doit être la première variable du fichier de configuration.

Stations ⟨*entier*⟩:⟨*entier*⟩ : donne la taille du réseau à étudier. Cette taille ne peut bien sûr pas être un intervalle si le réseau est de type Custom.

PraxiCars ⟨*entier*⟩:⟨*entier*⟩ : nombre de véhicules présents dans le système.

PraxiCarsRel ⟨*réel*⟩:⟨*réel*⟩ : nombre de véhicules, calculé comme une fraction du nombre critique. Ne peut pas être spécifié en même temps que PraxiCars.

Toutes ces variables sont obligatoires (à part PraxiCars et PraxiCarsMax qui ne sont mutuellement exclusives) et ne peuvent être utilisées qu'une fois. La description des caractéristiques du réseau est faite grâce aux variables ArrivalRate, Routing and Journey, qui donnent respectivement le taux d'arrivée aux stations, la matrice de routage et le temps de parcours entre les stations. Leur syntaxe dépend de la valeur de NetType :

Symmetrical (toutes les stations sont identiques)

- ArrivalRate est un intervalle de réels (mêmes arrivées partout),
- Routing ne peut pas être donné, puisque toutes les destinations sont équiprobables,
- Journey est un intervalle de réels.

Star (1 station centrale et Stations-1 stations périphériques)

- ArrivalRate est formé de deux intervalles de réels (station centrale et stations périphériques),
- Routing est un intervalle de réels donnant la probabilité de se rendre d'une station périphérique à la station centrale. Le reste de la matrice de routage en est déduit par symétrie,

Journey donne, sous forme de trois intervalles de réels, les durées moyennes de parcours cent. $\rightarrow$ périph., périph. $\rightarrow$ cent. et périph. $\rightarrow$ périph.

Custom (réseau quelconque)

ArrivalRate est un tableau de réels de taille Stations,

Routing est une matrice carrée de réels de taille Stations*Stations,

Journey est une matrice carrée de réels de taille Stations*Stations.

Il est possible, pour un réseau Custom, de remplacer ArrivalRate et Routing par une unique matrice Origine/Destination indiquant les nombre de trajets entre chaque paire de stations par unité de temps. La syntaxe à utiliser est

OrigDest $\langle$*réel*$\rangle$ $\langle$*réel*$\rangle$... , avec un nombre d'arguments exactement égal à Stations*Stations.

Ces deux manières de décrire la demande sont bien sûr exclusives.

Une dernière variable (facultative) décrivant le réseau est OverflowProba :

OverflowProba $\langle$*réel*$\rangle$: donne le taux maximum acceptable de dépassement de la taille des parkings. Ce nombre compris entre 0 et 1 est utilisé avec StationSize (voir plus loin).

4.4 Description des résultats demandés

Une fois le réseau Praxitèle défini, la deuxième partie de la configuration consiste à décrire les types de courbes désirées. On utilise pour cela les variables suivantes:

PlotVar $\langle$*donnée*$\rangle$ décrit la valeur placée en abscisse des courbes. La syntaxe de $\langle$*donnée*$\rangle$ est décrite plus bas.

Plot $\langle$*donnée*$\rangle$ crée une nouvelle courbe montrant l'évolution de la valeur $\langle$*donnée*$\rangle$,

Output $\langle$*donnée*$\rangle$ ajoûte une nouvelle donnée au fichier de sortie, sans l'ajouter au graphique. En utilisant uniquement OutputFile et Output, on peut créer un fichier de données ASCII, utilisable par la plupart des logiciels de traitement de données,

Title $\langle$*titre*$\rangle$ permet de donner un titre à l'élément après lequel il est placé.

- PlotFile : $\langle$*titre*$\rangle$ est placé au dessus du graphique,
- PlotVar : $\langle$*titre*$\rangle$ est placé sous l'axe des abscisses. Si il contient le caractère |, alors le texte placé après ce caractère sera place verticalement le long de l'axe des ordonnées,
- Plot : $\langle$*titre*$\rangle$ est placé en légende à l'intérieur du graphique. Si Title n'est pas utilisé, le titre par défaut sera $\langle$*donnée*$\rangle$.

Si $\langle$*titre*$\rangle$ contient une chaîne de la forme {$\langle$*donnée*$\rangle$}, cette chaîne sera remplacée par la valeur de $\langle$*donnée*$\rangle$. Toutefois, si $\langle$*donnée*$\rangle$ varie au cours des itérations, c'est sa dernière valeur qui apparaîtra.

Iterations $\langle$*entier*$\rangle$ déclare le nombre de points voulus dans les courbes. Le défaut est 30.

OutputFile $\langle$*nom*$\rangle$ permet de choisir le nom du fichier de données de sortie. Le défaut est `bignet.out`.

PlotFile $\langle$*nom*$\rangle$ permet de choisir le nom du fichier de commande GNU-Plot. Le défaut est `bignet.plt`.

Les valeur du paramètre $\langle$*donnée*$\rangle$ qui peut être utilisé avec PlotVar, Plot, Output et Title peuvent être réparties en deux classes :

1° Données globales

- Stations : le nombre de stations PRAXITÈLE ;
- PraxiCars : le nombre de véhicules ;
- PraxiCarsRel : la même chose, exprimée comme une fraction de PraxiCarsMax ;
- PraxiCarsMax : la valeur critique de de PraxiCars ;

- Usage : le taux d'utilisation des véhicules, exprimé en ⟨*voyages/véhicules/unité-de-temps*⟩ ;
- LossRate : le taux de perte de clients, c'est-à-dire la proportion de clients qui ne sont pas servis ;
- SatisRate : le taux de satisfaction, c'est-à-dire 1−LossRate ;
- StationSize : Donne la taille cumulée des parkings des stations permettant d'obtenir une probabilité de saturation des parkings inférieure à OverflowProba ;
- Bottleneck : la première station qui fait goulot d'étranglement sur le réseau ;
- BottleneckCount : le nombre de stations qui font goulot d'étranglement.

2º Données relatives à une station particulière : elles sont données sous la forme ⟨*nom*⟩[⟨*num*⟩], où ⟨*num*⟩ est un entier ou un nouveau paramètre ⟨*donnée*⟩ (pour l'instant, seul Bottleneck peut être utilisé ici).

- PraxiCars[⟨*num*⟩] : nombre moyen de véhicules présents dans la station ⟨*num*⟩ ;
- ArrivalRate[⟨*num*⟩] : taux d'arrivée de clients à la station ⟨*num*⟩ ;
- LossRate[⟨*num*⟩] : taux de perte à la station ⟨*num*⟩ ;
- SatisRate[⟨*num*⟩]] : le taux de satisfaction, c'est-à-dire 1−LossRate[⟨*num*⟩] ;
- StationSize[⟨*num*⟩] : Donne la taille du parking de la station ⟨*num*⟩ permettant d'obtenir une probabilité de saturation inférieure à OverflowProba.

4.5 Options de BigNet

En plus des variables contenues dans le fichier de configuration, il est possible de passer à `bignet.exe` des options sur la ligne de commande.

- ⟨*nom-fichier*⟩ : nom du fichier de configuration ;
- /v⟨*num*⟩ : indique la quantité de messages à produire lors de l'exécution. /v0 supprime tous les messages, /v1 affiche juste les

itérations, et les valeurs suivantes affichent des détails supplémentaires. La valeur par défaut est 0 sous Windows et 1 sous DOS ;

/l : écrit les messages dans le fichier `bignet.log` plutôt qu'à l'écran.

Toute autre option affiche la version et un court message d'aide.

4.6 Exemple complet de description de réseau

Pour conclure la description du programme, donnons les fichiers de configuration qui ont permis de produire les courbes de la section 3.3. Les données étant assez volumineuses, il est ici plus commode de les placer dans des fichiers séparés et d'utiliser ReadFile. Cette méthode est très pratique pour utiliser des fichiers de données de provenance extérieure.

Les fichiers présentés aux figures 4–6 utilisent de plus la possibilité de multiplier un ensemble de données par un coefficient fixé. Cela permet de ramener la matrice Origine-Destination en trajets par heures et d'exprimer les temps de parcours en heures (ils sont en minutes). De cette façon, toutes les durées sont exprimées en heures.

En plus des résultats nécessaires au graphique, on demande la sortie dans le fichier `paris.out` des taux de perte dans toutes les stations grace à la commande Output. Cela permet d'obtenir des données qui pourront être traitées par un programme quelconque.

Enfin, les titres utilisent la possibilité de la variable Title de remplacer certaines données par leur valeur, comme ici Stations, PraxiCarsMax et Bottleneck.

```
NetType Custom
Stations 7
PraxiCarsRel 0.5:3
Iterations 20

OrigDest*20 ReadFile m_trans.txt
Journey*60 ReadFile m_duree.txt

OutputFile paris.out
Output LossProba
Output LossProba[1]
Output LossProba[2]
Output LossProba[3]
Output LossProba[4]
Output LossProba[5]
Output LossProba[6]
Output LossProba[7]

PlotFile paris.plt
Title   "Réseau avec {Stations} stations"
PlotVar PraxiCars
Title   "Nombre de PraxiCars (critique={PraxiCarsMax})|Taux de perte"
Plot    LossProba Title "Total"
Plot LossProba[Bottleneck] Title "station critique : {Bottleneck}"
```

Fig. 4 – *Fichier principal de configuration*

```
187;328;0;354;38;527;222;
240;408;214;193;49;212;65;
94;0;314;163;195;0;0;
459;280;85;258;159;306;0;
393;132;264;52;352;364;189;
86;135;143;469;0;537;305;
277;296;0;171;111;358;231;
```

Fig. 5 – *Fichier de données annexe* `m_trans.txt`

```
4;4;8;8;12;7;8;
4;2;5;9;15;10;11;
8;5;6;6;12;16;19;
8;9;6;4;7;11;14;
12;15;12;7;5;7;15;
7;10;16;11;7;4;9;
8;11;19;14;15;9;4;
```

Fig. 6 – *Fichier de données annexe* `m_duree.txt`