File d'attente et chantier routier

Lecordier Alexis Candidat: 26969


VITESSE
Boul. Ville-Marie
Tunnel L.-H.-Lafontaine

Mise en contexte

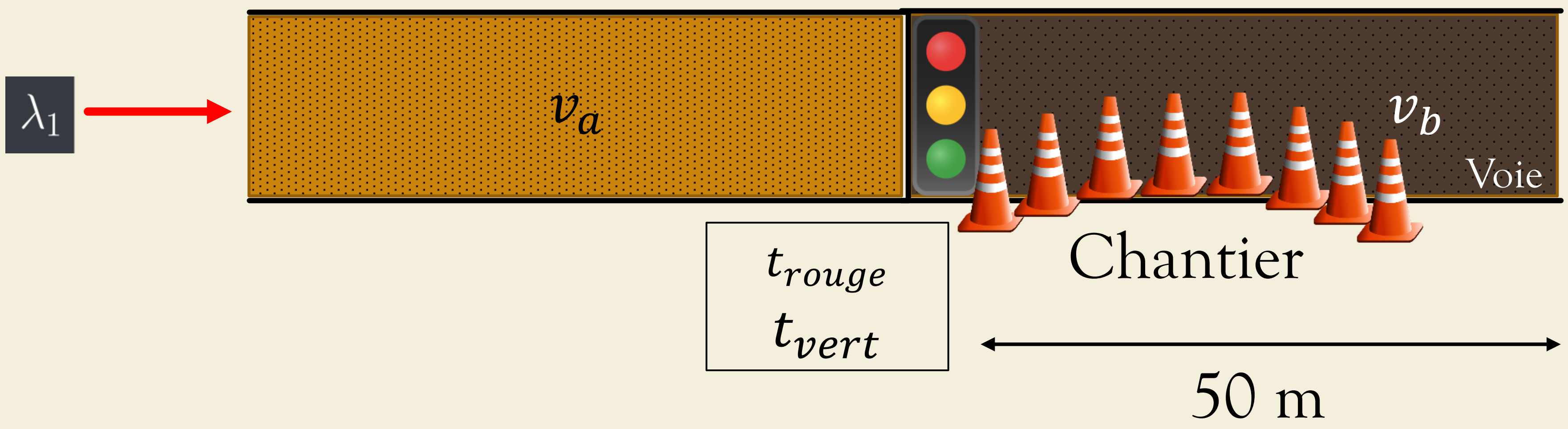


Fig. 1: Schéma du chantier routier

1er Approche

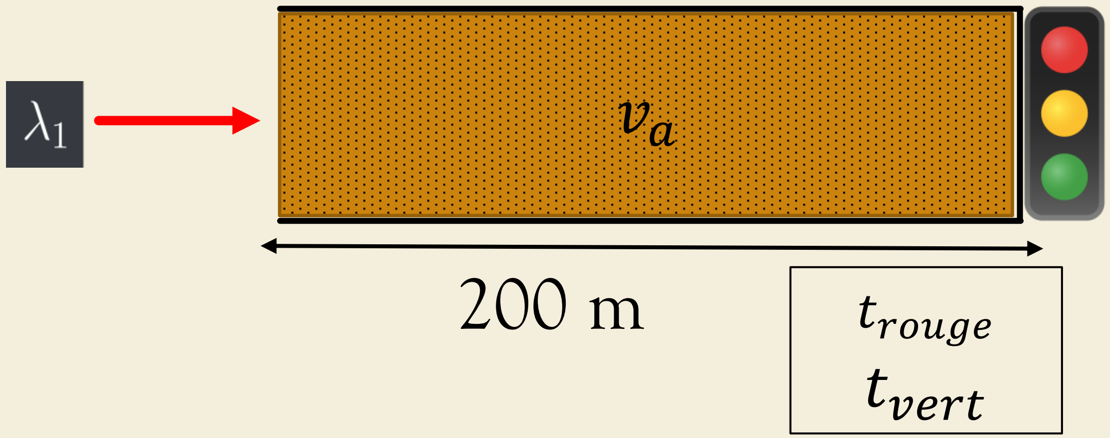


Fig.2: Route à une voie

V_{2n}: Nombre moyen de voitures restantes (après feu rouge)

V_{2n+1}: Nombre moyen de voitures restantes (après feu vert)

N_{2n+1}: Nombre moyen de voitures évacuées au feu vert (durant l'étape $2n+1$)

Evolution temporelle du système

On montre les relations:

$$\boxed{\begin{aligned} &V_{2n} = \lambda_1 t_{rouge} + V_{2n-1}\ , pour\ n > 0 \\ &V_{2n+1} = \lambda_1 t_{vert} + V_{2n} - N_{2n+1} \\ &N_{2n+1} = (t_{vert} - 5\frac{V_{2n}}{v_a})\frac{v_a}{200} \end{aligned}}$$

Evolution temporelle du système

Or la théorie des files d'attentes donne une condition de stabilité du système:

$\lambda_1 \leq \ \mu_n$ quelque soit n non nul.

DONC $\quad \frac{\lambda_1 t_v}{N_{2n+1}} \leq 1$

DONC (par croissance de (N_{2n+1})): La condition est $\boldsymbol{v_a \ \geq 200\ \lambda_1}$

Evolution temporelle du système

Sous cette hypothèse de stabilité:

$$l_{V_{2n}} = \lambda_1 t_{rouge} + l_{V_{2n+1}}$$

$$l_{V_{2n+1}} = \lambda_1 t_{vert} + l_{V_{2n}} - l_{N_{2n+1}}$$

$$l_{N_{2n+1}} = \left(t_{vert} - 5\frac{l_{V_{2n}}}{v_a}\right)\frac{v_a}{200}$$

Evolution temporelle du système

$$l_{V_{2n}} = \frac{1}{5}(v_a t_{vert} - 200\lambda_1(t_{vert} + t_{rouge}))$$
$$l_{V_{2n+1}} = \frac{1}{5}(v_a t_{vert} - 200\lambda_1(t_{vert} + t_{rouge})) - \lambda_{1 t_{rouge}}$$
$$l_{N_{2n+1}} = \lambda_1(t_{vert} + t_{rouge})$$

Evolution temporelle du système

Par positivité des limites:

$$t_{vert} \geq \lambda_1 \frac{205 t_{rouge}}{v_a - 200\lambda_1}$$

Puis, à cause de la taille de la voie:

$$t_{vert} \leq \frac{200(1 + \lambda_1 t_{rouge})}{v_a - 200\lambda_1}$$

Recherche d'une solution optimale

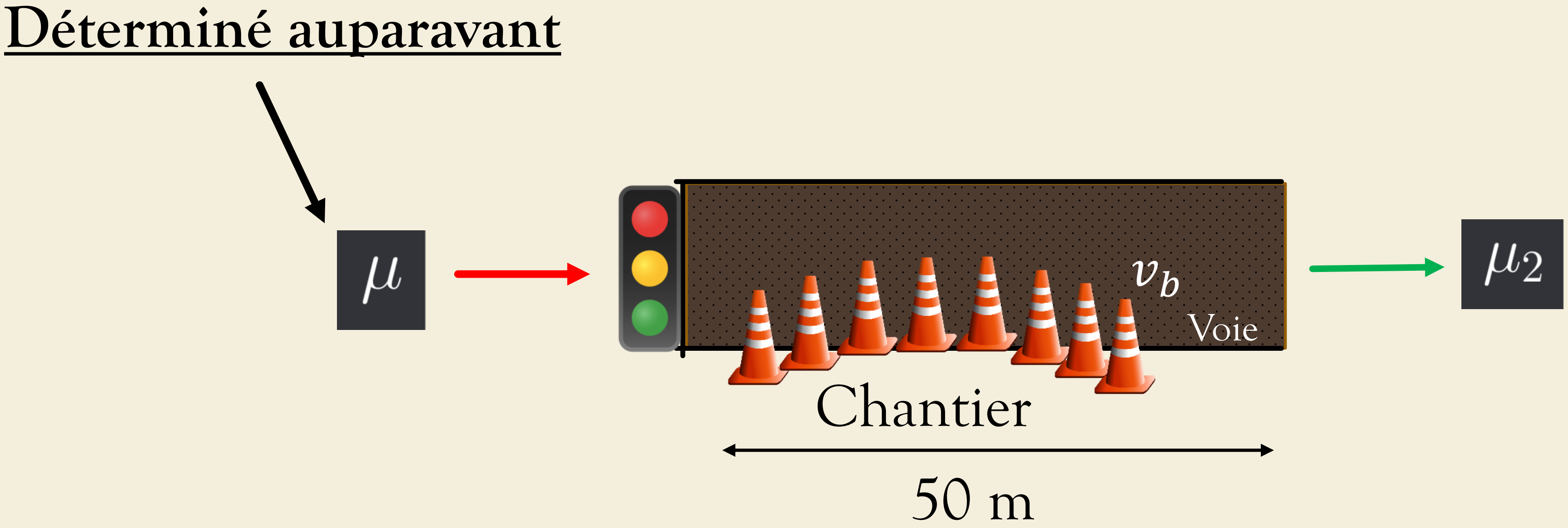


Fig. 3: Schéma du chantier routier (après feu)

Recherche d'une solution optimale

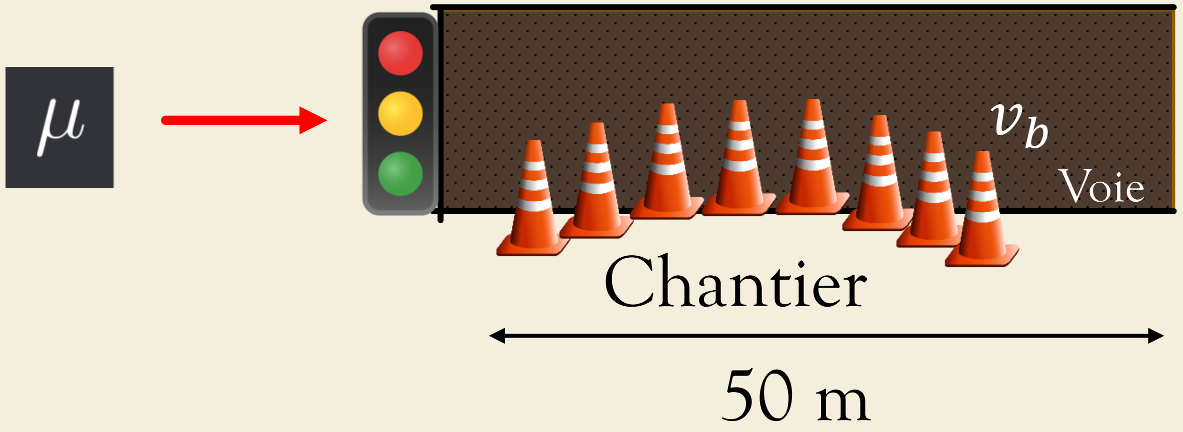


Fig. 3: Schéma du chantier routier avec fréquence μ

Condition de stabilité(D'après la théorie):

$$\mu = \frac{l_{N_{2n+1}}}{t_{vert}} < \mu_2$$

Or $\mu_2 = \frac{v_b}{50}$

Donc: $\boldsymbol{v_b} > 50\ \boldsymbol{\lambda_1}(1+ \frac{\boldsymbol{t_{rouge}}}{\boldsymbol{t_{vert}}})$

A MINIMISER

Cahier des charges/Contexte

v_a=90 km/h=25m/s> 200 λ_1 (λ_1=0,1 (*data.gouv.fr*))

$v_b < v_a$

Pas d'engorgement

$t_{vert} \geq 30$ s

$t_{rouge} \geq 30\ s$

Recherche optimale

$(v_a - 200\lambda_1)t_{vert} \leq 200(1 + \lambda_1 t_{rouge})$

$(v_a - 200\lambda_1)t_{vert} \geq \lambda_1 205 t_{rouge}$

$v_b > 50\ \lambda_1(1 + \frac{t_{rouge}}{t_{vert}})$

$20 > v_b$

$t_{vert} \geq 30$ s

Toutes les variables sont <u>positives</u> et entières λ_1=0,1 s-1 (*data.gouv.fr*)

Méthode du Simplexe Linéaire

Minimiser $Z = 50\lambda_1(\mathrm{t}_{\mathrm{rouge}} + \mathrm{t}_{\mathrm{vert}})$

$-200 = (200\lambda_1 - v_a)t_{vert} + \lambda_1 t_{rouge}) - \mathrm{X}_1$

$0 = +(v_a - 200\lambda_1)t_{vert} - 205\lambda_1 t_{rouge} - \mathrm{X}_2$

$0 = -t_{vert} + t_{rouge} - \mathrm{X}_3$

$30 = t_{vert} - \mathrm{X}_4$

$\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3, \mathrm{X}_4 \geq 0$

Toutes les variables sont <u>positives</u> et entières λ_1=0,1 s-1 (*data.gouv.fr*)

Programmation

Le code en Annexe donne comme solution optimale:

```
>>> L=0.1

>>> (executing lines 1 to 47 of "Tipe Chantier routier.py")
Valeur optimale de Z : 299.9999999896189
Variables optimales : [3.00000000e+01 3.00000000e+01 2.65800354e-01
2.65800354e-01
 3.04489073e+01 4.66746641e-11 0.00000000e+00 6.24734200e+02]
```

Donc t_{vert}=30s= t_{rouge} est une solution optimale.

On a aussi Z=300, donc: $\boldsymbol{v_b>10}$

Réponse au cahier des charges

$\boldsymbol{v_b}$=11 m/s=36 km/h < $\boldsymbol{v_a}$

$t_{vert} \geq 30\ \mathrm{s}$

$t_{rouge} \geq 30\ s$

Pas d'engorgement

Arrivée variable de clients

On suppose ici que λ_1 évolue au cours de la journée

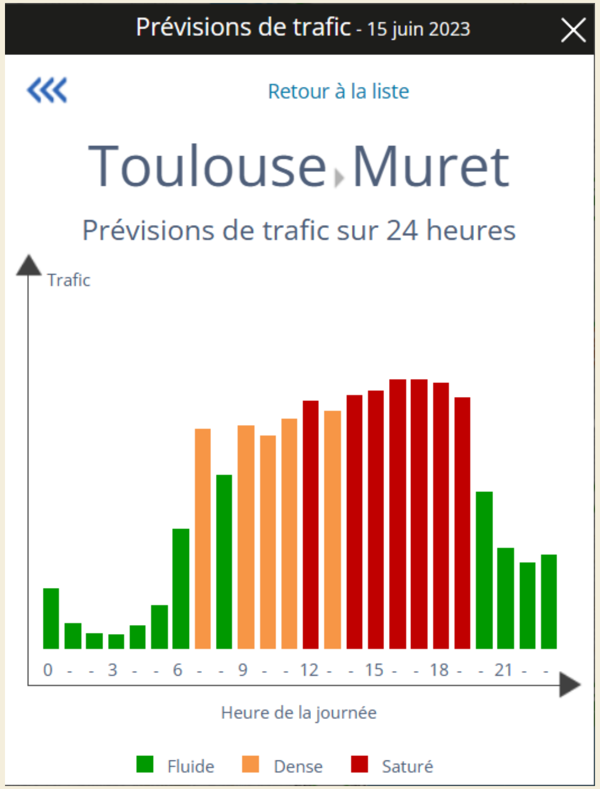
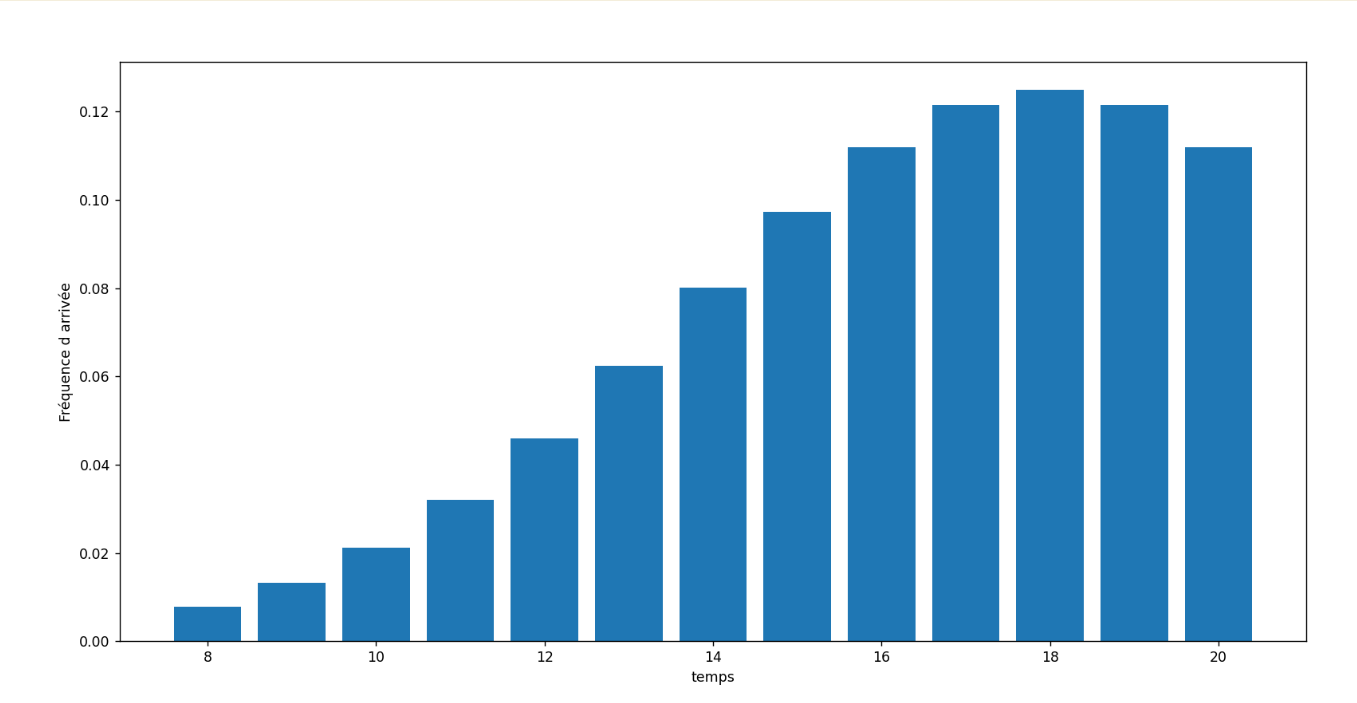


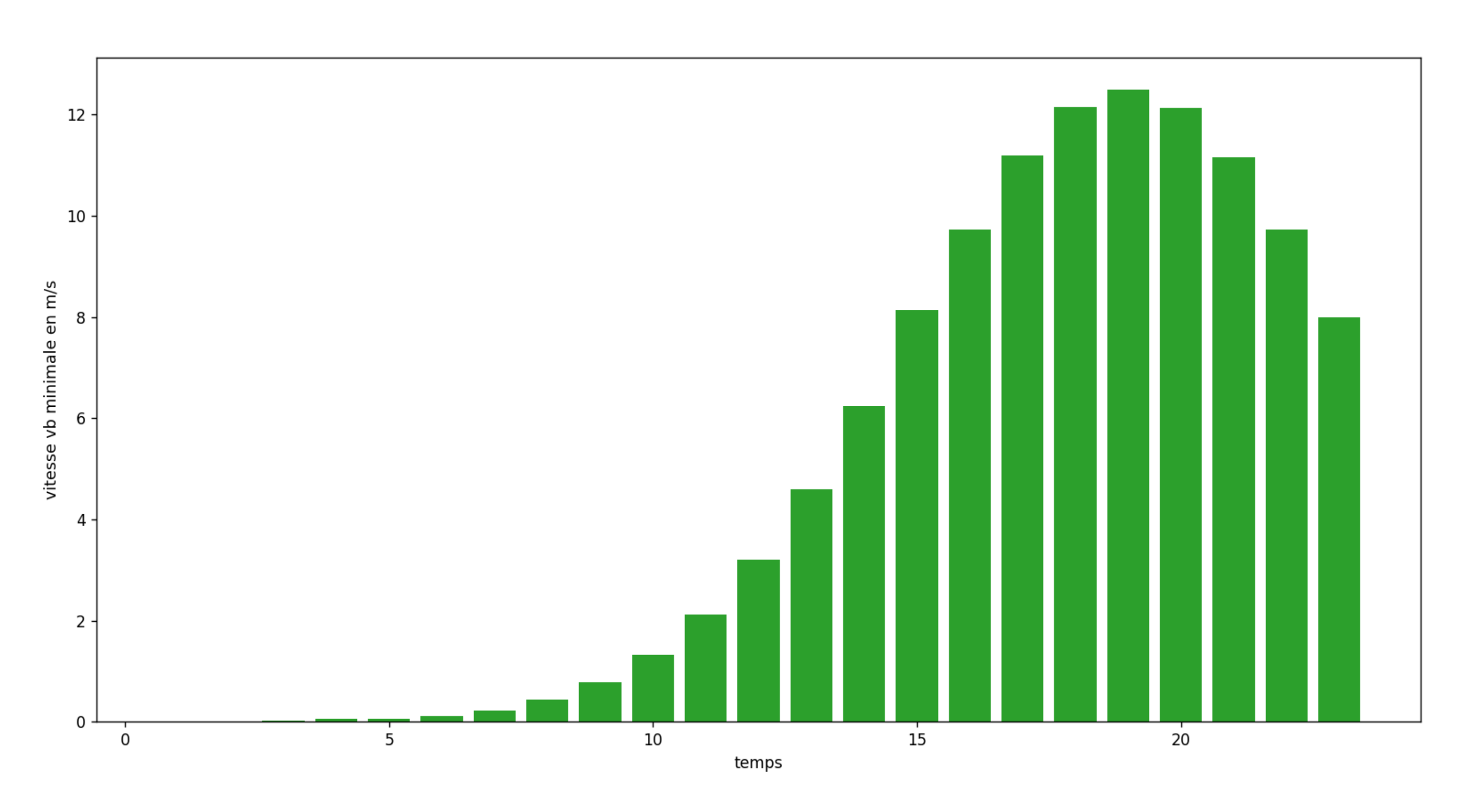
Fig 4- Vinci-Autoroute.com

Or $v_a \geq 200\lambda_1$

Donc $\lambda_1 \leq 25/200$



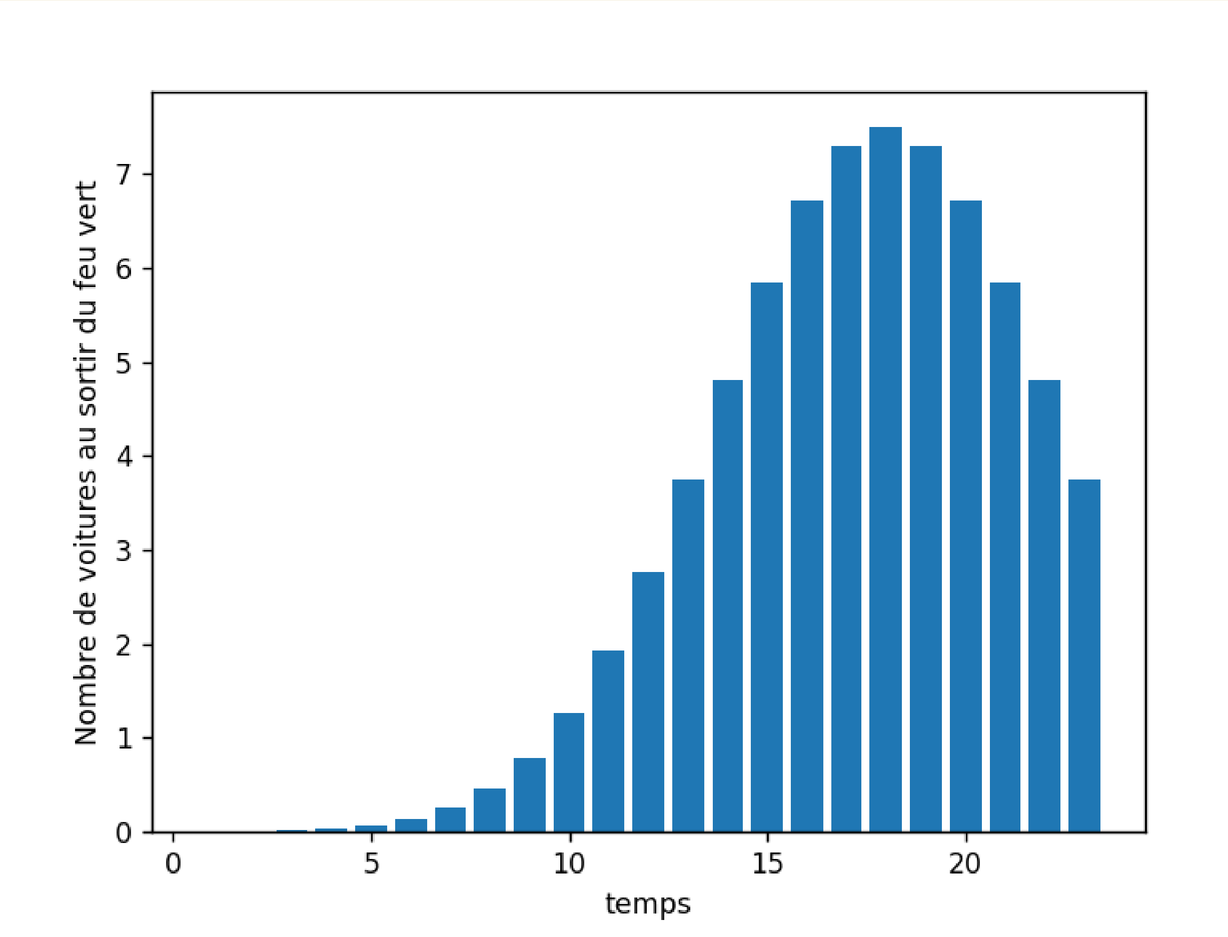
Arrivée variable de clients- Réponse



ET $t_{vert}=30=t_{rouge}$ quelque soit le temps.

Arrivée variable de clients- Réponse

On peut même prédire le nombre moyen de véhicules évacués au feu vert



$$l_{N_{2n+1}} = \lambda_1(t_{vert} + t_{rouge})$$

Conclusion

La stratégie optimale consiste à faire varier la vitesse sur le chantier de manière à éviter les bouchons. Or on a trouvé comment faire varier cette vitesse.

Limites du Modèle

Pas de temps de freinage
Pas de temps de démarrage
Pas de poids lourds
Chantier étudié en régime stationnaire
Pas de véhicule en panne ...

```
from scipy.optimize import minimize

def objectif(x):
    return 50 * L * (x[0] + x[1])

def contrainte1(x):
    return 200 * L * x[0] - x[2] - x[7] + 25

def contrainte2(x):
    return 200 * L * x[1] - 205 * L * x[2] - x[3] + x[4] - x[7]

def contrainte3(x):
    return -x[2] + x[3] - x[5]

def contrainte4(x):
    return x[0] - 30

def contrainte5(x):
    return x[1] - 30

def contrainte6(x):
    return x[0] - x[1]

def contrainte7(x):
    return x[2] - L
```

```python
# Conditions initiales
x0 = [0, 0, 0, 0, 0, 0, 0, 0]

# Contraintes
contraintes = [{'type': 'eq', 'fun': contrainte1},
               {'type': 'eq', 'fun': contrainte2},
               {'type': 'eq', 'fun': contrainte3},
               {'type': 'ineq', 'fun': contrainte4},
               {'type': 'ineq', 'fun': contrainte5},
               {'type': 'ineq', 'fun': contrainte6},
               {'type': 'ineq', 'fun': contrainte7}]

# Contraintes de non-négativité
bornes = [(0, None), (0, None), (0, None), (0, None), (0, None), (0, None), (0, None), (0, None)]

# Appel à la fonction minimize
resultat = minimize(objectif, x0, method='SLSQP', bounds=bornes, constraints=contraintes)

# Affichage des résultats
print("Valeur optimale de Z :", resultat.fun)
print("Variables optimales :", resultat.x)
```

```python
y1=[]

import numpy as np
import matplotlib.pyplot as plt

# Définition de la fonction gaussienne étalée
# Paramètres de la gaussienne étalée
mu = 18  # Moyenne
sigma = 6  # Écart-type
def f(x, mu, sigma):
    return (25/200)*np.exp(-((x-mu)/sigma)**2)

# Génération des valeurs de x
x = np.arange(8,21)

# Paramètres de la gaussienne étalée
mu = 18  # Moyenne
sigma = 6  # Écart-type

# Calcul des valeurs de y correspondantes
y = f(x, mu, sigma)

# Création du graphique en barres
plt.bar(x, y)

# Définition des labels des axes
plt.xlabel('temps')
plt.ylabel('Fréquence d arrivée')

```

```python
# Affichage du graphique
plt.show()

x_1 = np.arange(1,24)

print(len(x_1))
plt.bar(x_1, y1)
plt.xlabel('temps')
plt.ylabel('vitesse vb minimale en m/s')
plt.show()
##Nombre moyen de clients au sortir du feu vert
y2=[f(x, mu, sigma)*60 for x in range(1,24)]
plt.bar(x_1, y2)
plt.xlabel('temps')
plt.ylabel('Nombre de voitures au sortir du feu vert')

plt.show()
```