INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING
DOI: 10.46300/9106.2021.15.172 Volume 15, 2021

# Optimization of Transport Flow on Two Paths with Respect to the Passengers Time Costs

Paolo Mercorelli
Institute of Product and Process Innovation
Leuphana University of Lueneburg
D-21335 Lueneburg, Universitaetsallee 1
Germany
Tel. +49.4131.677-1896
mercorelli@uni.leuphana.de

Received: April 9, 2021. Revised: September 28, 2021. Accepted: October 9, 2021. Published: October 29, 2021.

**Abstract: Transportation of passengers by public transport is one of the most important tasks of the economy. Waste of passengers' time plays a special role in optimizing the movement of vehicles on urban routes. This paper considers an optimization problem in the context of two paths and proposes an optimal strategy of transport with respect to the costs of the travel. The optimization method is based on the Newton Method and the software is realized using Python.**

**Key–Words: Optimization, Gradient Methods, Traffic Control**

## I. INTRODUCTION

One aspect of the Smart City concept, which is currently being launched everywhere, is a better control of the flow of traffic. Mobility is to be increased, switching between modes of transport promoted, congestion avoided, air pollution reduced and the use of cars reduced through a better organization of public transport and with sharing models or, at some point, autonomous vehicles. Above all, Smart City is a concept of optimization that underlies the digitization and evaluation of data on a large scale. In a pilot project in Newcastle, UK, the optimization of the flow of traffic is now being tested using 20 smart traffic lights. The communication system in the vehicles is directly connected to the Traffic Control Center (UTMC). This makes it possible for the traffic lights, as they say, to ”speak” to the motorists, which is, however, a somewhat skewed humanization of the situation. The vehicles taking part in the experiment are recognized 100 meters away from the traffic light and given priority by the traffic light switch, which turns green when the traffic flow allows it. In addition, the driver receives ”personalized” information about traffic disruptions and information about the speed at which he should drive in order to get a green phase at the next traffic lights. This is not really new, the preference for public transport is already practiced, traffic reports in real time are also available in navigation systems and the analysis of data on a large scale is the basis. In any case, transportation of passengers by public transport is one of the most important tasks of the economy. Waste of passengers' time plays a special role in optimizing the movement of vehicles on urban routes. The main indicator here is the loss of passenger hour. Therefore, when optimizing the work of urban passenger transport not only transport costs must be considered, but also the socio-economic aspect associated with downtime passenger at stopping points. According to Antoshvili M. et al. in [1], sociological researches demonstrate that reducing the passenger's travel time from home to work by 10 minutes leads to an increase in labour productivity by approximately 4%. Moreover, if the speed of public transport increases, then the carrying capacity and performance rise. With an increase in the interval of movement of vehicles along this route, the time spent by passengers increases, but transport costs decrease and vice versa. A compromise is needed between the socio-economic importance of passenger transportation and transport costs (we restrict ourselves to two routes).

## II. THEORETHICAL FOUNDATION

### *A. Queueing theory*

Queueing theory is an applied mathematical discipline dealing with the performance of technical systems, which, in what follows, are referred to as queueing systems for processing of flows of customers

(Bocharov P.P. et al.), see [2].

We can extract the following components that are common to all queueing systems:

- ***input flow*** of customers, that is, the process of arrival of the customers at the systems;
- ***system structure***, that is, number and types of the servers, as well as the capacities of the buffers located before all and/or individual servers;
- ***times of customer service*** by the servers, that is, the actual times (disregarding the times of waiting and service interrupt) during which the customers undergo complete service and depart from the systems;
- ***service discipline***, that is, the process of allocating the customers to the servers, generating the queues, picking the customers from the queues, and so on (Bocharov P.P. et al.) see [2].

### *B. Newton's method*

Newton's idea is simple: to find the root of a function, find the tangent line of the function at your current guess for the root, then your next guess will be where the tangent line intersects the x-axis. If the function is linear, then the root will be found in just one step, and if the function is close to linear, then this will give a very good approximation to the root. To run the algorithm, we need an initial guess, and the better it is, the better the algorithm will perform.

If we know the previous guess $x_k$ , Newton's idea allows us to explicitly find $x_{k+1}$. The tangent line to $f(x)$ at $x_k$ is given by

$$y - f(x_k) = f(x_k)(x - x_k). \tag{1}$$

Newton's next iteration is defined where this line intersects the x-axis, and we call this point $(x_{k+1}, 0)$. Inserting this point into the tangent line gives

$$0 - f(x_k) = f(x_k)(x_{k+1} - x_k). \tag{2}$$

Now solve for $x_{k+1}$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{3}$$

We start the process with an arbitrary initial value $x_0$. The method usually converges, if this initial guess is close enough to the unknown zero, and that $f'(x_0) = 0$.

Newton's method can run into one of several difficulties:

1. If the initial guess is not sufficiently close, the initial tangent line approximation could be terrible and prevent convergence.
2. Vanishing derivatives. If $f'(x_k) \approx 0$, the tangent line has no root, or one that is very far away. Here, Newton's method can become numerically unstable and may not converge.
3. Cycles. You can construct examples where Newton's method jumps between two points (so $x_{k+2} = x_k$) and never converges (T. Heister et al., 2019), see [3].

In summary, Newton's method is not guaranteed to work, but when it does, it is very fast.

## III. Formulation of the Problem

To formulate the problem, consider the next situation. We have two routes, which serve three passenger flows:

1. Passenger flow is transported by public transport only of the first path;
2. Passenger flow is transported by public transport only of the second path;
3. Passenger flow is transported by public transport of two ways together.

On both routes, people wait for the bus at the bus stop. The bus arrives at the stop and picks up passengers. Some time passes, and people appear at the bus stop again. The situation repeats. Again, some time passes, and the situation repeats. Thus, we need to have a certain number of buses per unit of time to serve passenger flows. In addition, transporting passengers requires transportation costs, and waiting for a bus by passengers has a certain price. If there is information about the cost of one transportation per path and the cost passenger per hour, the task of finding the optimal intervals of transport movement along two ways can be set.

We denote:

$\lambda_1$ , $\lambda_2$ , $\lambda_0$ the intensity of the previously mentioned passenger flows;

$\mu_1$ , $\mu_2$ the required intensity of the Poisson flow of transports along the corresponding way per unit of time;

$\gamma$ the cost of a passenger hour;

$\alpha_1$ , $\alpha_2$ the cost of one transportation one way.

If transport flows are Poisson, independent from each other's and passenger flows, then the share of passenger flow, which is transported by each path, is

proportional to its traffic intensity. Therefore average passenger time losses per unit time on the corresponding transport way are:

$$\frac{\lambda_1}{\mu_1}+\frac{\lambda_0}{\mu_1+\mu_2} \tag{4}$$

$$\frac{\lambda_2}{\mu_2}+\frac{\lambda_0}{\mu_1+\mu_2}. \tag{5}$$

Total passenger losses due to waiting for public transport:

$$h(\mu)=\gamma\left(\frac{\lambda_1}{\mu_1}+\frac{\lambda_2}{\mu_2}+\frac{\lambda_0}{\mu_1+\mu_2}\right). \tag{6}$$

Transportation costs for travel:

$$g(\mu)=\alpha_1\mu_1+\alpha_2\mu_2. \tag{7}$$

Objective function (costs of passengers and transport):

$$f(\mu)=\gamma\left(\frac{\lambda_1}{\mu_1}+\frac{\lambda_2}{\mu_2}+\frac{\lambda_0}{\mu_1+\mu_2}\right)+\alpha_1\mu_1+\alpha_2\mu_2\to\min_{\mu}. \tag{8}$$

We calculate optimal traffic intervals given that

$$\alpha_1=\alpha_2=\alpha. \tag{9}$$

We can show that we have one required minimum and that we have the following equations at the optimal point:

$$\gamma\left(-\frac{\lambda_1}{{\mu_1}^2}-\frac{\lambda_0}{(\mu_1+\mu_2)^2}\right)+\alpha=0. \tag{10}$$

$$\gamma\left(-\frac{\lambda_2}{{\mu_2}^2}-\frac{\lambda_0}{(\mu_1+\mu_2)^2}\right)+\alpha=0, \tag{11}$$

whence we have

$$\frac{\lambda_1}{{\mu_1}^2}=\frac{\lambda_2}{{\mu_2}^2}. \tag{12}$$

Expressing $\mu_2$ from (12) and substituting to (10) , we get

$$\gamma\left[-\frac{\lambda_1}{{\mu_1}^2}-\frac{\lambda_0}{{\mu_1}^2\left(1+\sqrt{\frac{\lambda_1}{\lambda_2}}\right)^2}\right]+\alpha=0. \tag{13}$$

Thus, optimal solution is

$$\mu_1=\sqrt{\frac{\gamma\lambda_1(\lambda_1+\lambda_2+2\sqrt{\lambda_1\lambda_2}+\lambda_0)}{\alpha(\lambda_1+\lambda_2+2\sqrt{\lambda_1\lambda_2}}} \tag{14}$$

$$\mu_2=\sqrt{\frac{\gamma\lambda_2(\lambda_1+\lambda_2+2\sqrt{\lambda_1\lambda_2}+\lambda_0)}{\alpha(\lambda_1+\lambda_2+2\sqrt{\lambda_1\lambda_2}}}. \tag{15}$$

In case of violation of (9), we do not have the optimal traffic intervals in an analytical form. A numerical solution is needed, and for this, Newton's method is used.

Considering that the Newton's iterative algorithm has the following form:

$${\mu_1}^{k+1}={\mu_1}^k-\frac{f_1({\mu_1}^k,{\mu_2}^k)*f_{2,2}({\mu_1}^k,{\mu_2}^k)-f_2({\mu_1}^k,{\mu_2}^k)*f_{1,2}({\mu_1}^k,{\mu_2}^k)}{D({\mu_1}^k,{\mu_2}^k)} \tag{16}$$

$${\mu_2}^{k+1}={\mu_2}^k-\frac{f_2({\mu_1}^k,{\mu_2}^k)*f_{1,1}({\mu_1}^k,{\mu_2}^k)-f_1({\mu_1}^k,{\mu_2}^k)*f_{1,2}({\mu_1}^k,{\mu_2}^k)}{D({\mu_1}^k,{\mu_2}^k)} \tag{17}$$

where,

$f_1$ the first derivative of $f(\mu)$ with respect to the variable $\mu_1$;

$f_2$ the first derivative of $f(\mu)$ with respect to the variable $\mu_2$;

$f_{1,1}$ the second derivative of $f(\mu)$ with respect to the variable $\mu_1$;

$f_{2,2}$ the second derivative of $f(\mu)$ with respect to the variable $\mu_2$;

$f_{1,2}$ the derivative of $f_1$ with respect to the variable $\mu_2$;

$D$ the determinant of the matrix of second derivatives;

$k$ the number of iteration.

As example, we consider a problem with the following parameter values. The passenger flows are $\lambda_1=100$ , $\lambda_2=50$ , $\lambda_0=150$ people per hour. The transport costs on routes are $\alpha_1=115$ and $\alpha_2=92$ euro. The passenger time cost is $\gamma=5,7$ euro per hour. We use Python to implement this problem and

find a solution.

## IV. Results

Figure 1 shows the objective function on the plane of two variables.

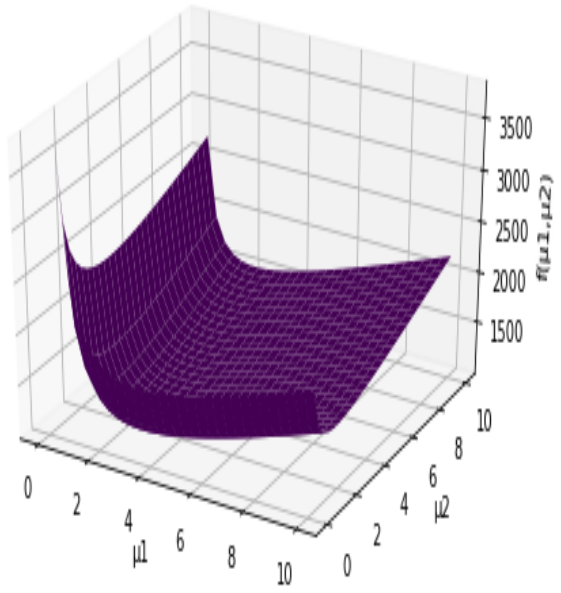


Figure 1: Total costs of transport and passengers

It can be seen from Fig. 1 that our function has a convex point. This indicates the presence of a minimum of the function and the possibility of rapid convergence of Newton's method.

The process of iterating over Newton's method is shown in the Table 1. Having performed calculations by Newton's method, using for obtaining the optimal traffic intensity of vehicles, high convergence of the method can be noted.

Under these conditions, the expected optimal transport costs are 513,9 euro per hour at traffic intensities $\mu_1 \approx 2,6743$ and $\mu_2 \approx 2,2430$. The average number of transported passengers is 67,9 for the first and 52,8 for the second routes per travel. If the fare is 2,1 euro, the profit on the first route will be 27,59 euro, and on the second one is 18,87 euro per trip.

## V. CONCLUSION

When organizing work, urban passenger transport needs accounting not only transport costs, but also socio-economic importance of transportation. A particular role in optimizing the movement of vehicles on city routes is played by loss of passengers' time.

This paper proposes an optimization problem in the context of two paths and develops an optimal strategy of transport with respect to the costs of the travel. The optimization method is based on the Newton Method and the software to optimize is realized using Python.

## Acknowledgement

Many thanks are addressed to Aleksandra Viktorowa for her essential contribution to this work. Without her contribution this work would not have been accomplished.

## References

[1] Antoshvili, M. Optimizatsiya gorodskikh avtobusnykh perevozok, M.E. Antoshvili, C. Yu. Liberman, I.V. Spirin. M.: Transport, 1985.

[2] Bocharov, P. Queueing Theory, P.P. Bocharov, C. D'Apice, A.V. Pechinkin, S.Salerno. VSP, Utrecht, Boston, 2004.

[3] Heister, T. Numerical Analysis. An introduction, Tino Hiester, Leo G. Rebholz, Fei Xue. Walter de Gruyter GmbH, Berlin/Boston, 2019.

### Appendix

The Python code.

```
import matplotlib.pyplot as plt
import numpy as np
from sympy import *
from mpl_toolkits import mplot3d

# Parameters
Pass_flow_0 = 150
Pass_flow_1 = 100
Pass_flow_2 = 50
Alpha_1 = 115
Alpha_2 = 92
Gamma = 5.7
Fare = 2.1

# The required parameters
Mu_1 = Symbol('Mu_1')
Mu_2 = Symbol('Mu_2')

#Cost functions
# Transport costs
g_mu = Alpha_1*Mu_1 + Alpha_2*Mu_2
# Passenger's cost
h_mu = Gamma*(Pass_flow_1/Mu_1...
+ Pass_flow_2/Mu_2 + ...
    Pass_flow_0/(Mu_1 + Mu_2))
# Objective function
```

Table 1: Newton's method implementation

| $k$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $\mu_1$ | 1 | 1,4306 | 1,9424 | 2,4031 | 2,6358 | 2,6743 |
| $\mu_2$ | 1 | 1,4015 | 1,8307 | 2,1388 | 2,2369 | 2,2430 |

```
f_mu = g_mu+ h_mu

#Plot of function
func = lambdify([Mu_1,Mu_2],f_mu)
x =np.linspace(0,10,30)
y = np.linspace(0,10,30)
X,Y = np.meshgrid(x,y)
Z = func(X,Y)
fig = plt.figure(figsize=(8,4))
ax = plt.axes(projection = '3d')
ax.plot_surface(X,Y,Z,rstride=1,cstride=1,
cmap='viridis',edgecolor='none')
ax.set_xlabel(" 1 ")
ax.set_ylabel(" 2 ")
ax.set_zlabel("f( 1 , 2 )")

# Derivative of function
fun_1 = f_mu.diff(Mu_1)
fun_11 = fun_1.diff(Mu_1)
fun_2 = f_mu.diff(Mu_2)
fun_22 = fun_2.diff(Mu_2)
fun_12 = fun_1.diff(Mu_2)

# Derivative of function in one point
f_1 = lambdify([Mu_1,Mu_2],fun_1)
f_11 = lambdify([Mu_1,Mu_2],fun_11)
f_2 = lambdify([Mu_1,Mu_2],fun_2)
f_22 = lambdify([Mu_1,Mu_2],fun_22)
f_12 = lambdify([Mu_1,Mu_2],fun_12)

# Iteration parameters
k = 0
Mu_10 = 1
Mu_20 =1

# First step of algorithm
b=np.array([[f_11(Mu_10,Mu_20),..
f_12(Mu_10,Mu_20)],.
[f_12(Mu_10,Mu_20),f_22(Mu_10,Mu_20)]])
D = np.linalg.det(b)
d1=(f_1(Mu_10,Mu_20)*f_22(Mu_10,Mu_20)..
-f_2(Mu_10,Mu_20)*f_12(Mu_10,Mu_20))/D
d2 = ...
    (f_2(Mu_10,Mu_20)*f_11(Mu_10,Mu_20)..
-f_1(Mu_10,Mu_20)*f_12(Mu_10,Mu_20))/D
p=[]

# Newton's algorithm
while abs(d1)>1e-4 and abs(d2)>1e-4:
Mu_11 = Mu_10 - d1
Mu_21 = Mu_20 - d2
k = k+1
Mu_10 = Mu_11
Mu_20 = Mu_21
b=np.array([[f_11(Mu_10,Mu_20),...
f_12(Mu_10,Mu_20)],..
[f_12(Mu_10,Mu_20),f_22(Mu_10,Mu_20)]])
D=np.linalg.det(b)
d1=(f_1(Mu_10,Mu_20)*f_22(Mu_10,Mu_20)..
-f_2(Mu_10,Mu_20)*f_12(Mu_10,Mu_20))/D
d2 = ...
    (f_2(Mu_10,Mu_20)*f_11(Mu_10,Mu_20)..
-f_1(Mu_10,Mu_20)*f_12(Mu_10,Mu_20))/D
p.append([Mu_10,Mu_20,k])

# Value of required parameters
Mu_10, Mu_20

#Iterations
p

# Optimal transport costs
g = lambdify([Mu_1,Mu_2],g_mu)
g(Mu_10,Mu_20)

# Average number of transported ...
    passenger
avg_1 = Pass_flow_1/Mu_10 + ...
    Pass_flow_0/(Mu_10+Mu_20)
avg_2 = Pass_flow_2/Mu_20 + ...
    Pass_flow_0/(Mu_10+Mu_20)
avg_1, avg_2

#Profit from routes
pr_1 = avg_1*Fare - Alpha_1
pr_2 = avg_2*Fare - Alpha_2
pr_1,pr_2
```