

Optimization Model of the Buses Number on the Route Based on Queueing Theory in a Smart City

Liliia Podlesna¹, Myroslava Bublyk²[0000-0003-2403-0784], Igor Grybyk³[0000-0002-8726-509X],
Yurii Matseliukh⁴[0000-0002-1721-7703], Yevhen Burov⁵[0000-0001-8653-1520], Petro Kravets⁶[0000-
0001-8569-423X], Olga Lozynska⁷[0000-0003-2882-3546], Ihor Karpov⁸[0000-0003-4885-5078], Ivan
Peleshchak⁹[0000-0002-7481-8628], and Roman Peleshchak¹⁰[0000-0002-0536-3252]

¹⁻⁹Lviv Polytechnic National University, Lviv, Ukraine

¹⁰Ivan Franko Drogobych State Pedagogical University, Drogobych, Ukraine
liliia.podlesna.sa.2017@lpnu.ua¹, my.bublyk@gmail.com²,
gribik@ukr.net³, indeed.post@gmail.com⁴

Abstract. The busload on the route is modeled, the main statistical characteristics of the process on route number 10 were calculated, and the optimal number of buses on the route is calculated, which would maximize the total number of passengers per flight. The developed program allows us to calculate the optimal number of buses on the route according to the previously defined criteria in a certain period of the day on weekdays and weekends (including holidays). Diagrams of distribution of passengers and their dependence on the number of buses and their schedules are given in the work.

Keywords: Transport, Bus Route, Smart City, Queueing Theory, Optimization Model.

1 Introduction

A promising direction for improving urban transport systems is to determine the optimal number of vehicles on the route. The solution to this problem involves bringing to the route so many large or medium-sized buses that can provide comfortable conditions for passengers and fully meet the demand during rush hour [1-7]. Among the programs we know, most of them help passengers see the location of the vehicle, make a route to get from one part of the city to another, and so on. There are useful applications for tracking traffic, which count the number of buses on the route and indicate where they are located too. This proves the actuality of developing a program that would be analyzed at different times of the day and different days of the week the number of passengers who want to take the bus. Currently, there are no analogs of such a program, which would analyze the data on the number of buses on the route for different parameters and justify how many and which buses on the route are the most optimal. Potential consumers of the developed software are both civil servants (heads of cities and relevant government agencies) and heads of private transport companies.

Copyright © 2020 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2 System Analysis

To develop the system and understand the essence of the task, a plan of its work was drawn up [8-15]. At the stages of system development, certain independent tasks were divided into works, which helped to accelerate the implementation of the system [16-23].

The use case diagram was built for the designed system (Fig. 1). It was singled out two main actors: the program creator and the bus driver. The main goal of the program creator is to optimize the number of buses on the flight along a certain route. To do this, the program creator needs to collect data on the number of passengers on different days, as well as different hours during these days, to statistically present this data. This will allow the program creator to determine the optimal number of buses for a given day and time.

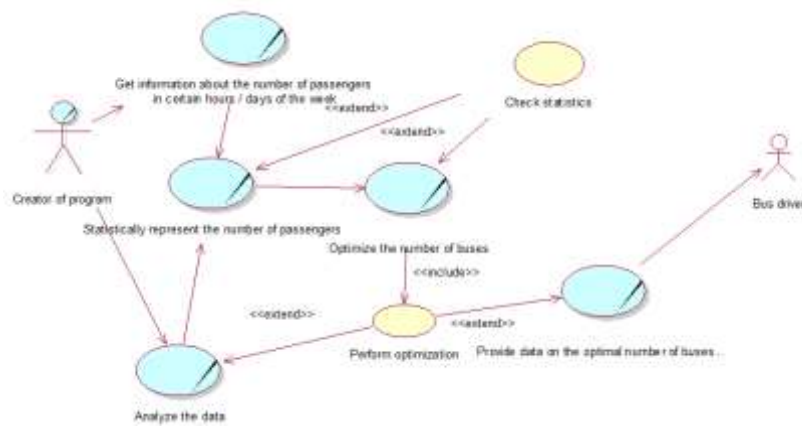


Fig. 1. Use case diagram

To build a class diagram of this system (fig.2), you should add the rest of the classes and associations, as well as clarify the stereotypes, attributes and operations of these classes [24-32]. Operations have been added for the Transport Company class: check the correctness of entering personal data, establish the user's identity, check the number of passengers, and optimize the number of buses (data identifier: Integer). An operation has been added for the Company Bus class: go on a flight (). Operations have been added for the User Interface class: show bus usage statistics, show a page for entering personal data, and specify the type of transport.

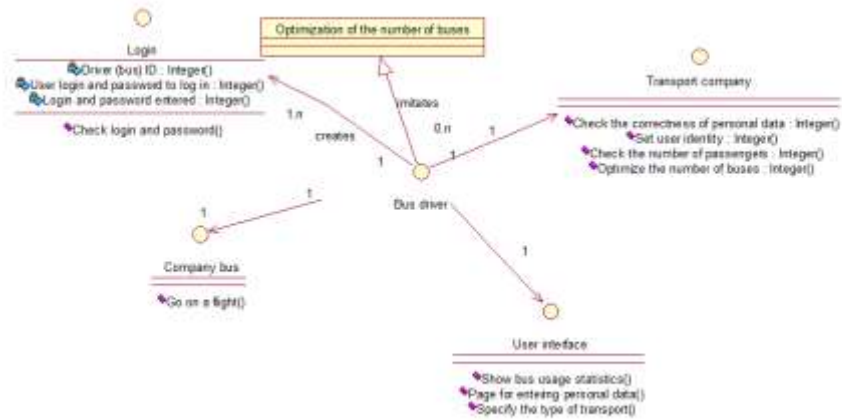


Fig. 2. Class diagram

Fig. 3 presents a cooperation diagram, which shows not only the sequence of interaction but also all the structural relationships between the objects involved in this interaction. In contrast to a sequence diagram, a cooperation diagram shows only the relationship between objects that play certain roles in the interaction [24-32].

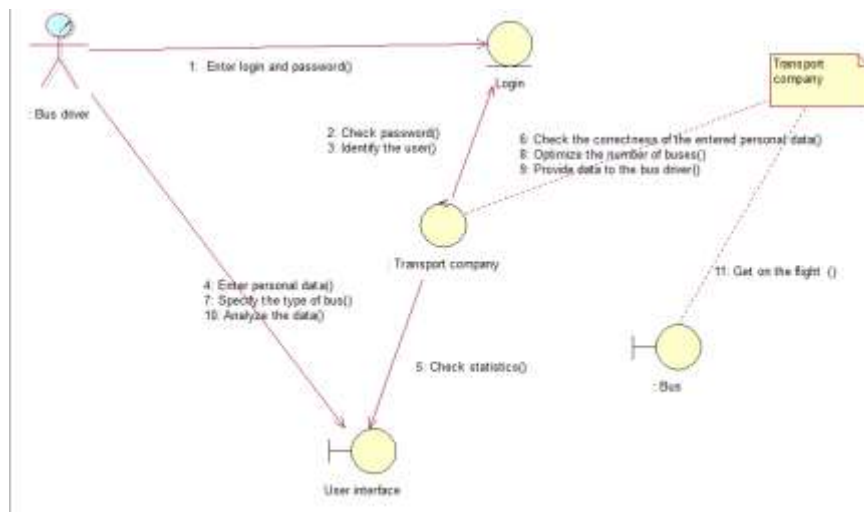


Fig. 3. Cooperation diagram

Fig. 4 shows a state diagram. This type of diagram is used to explain how complex work objects. The state diagram shows how an object moves from one state to another.

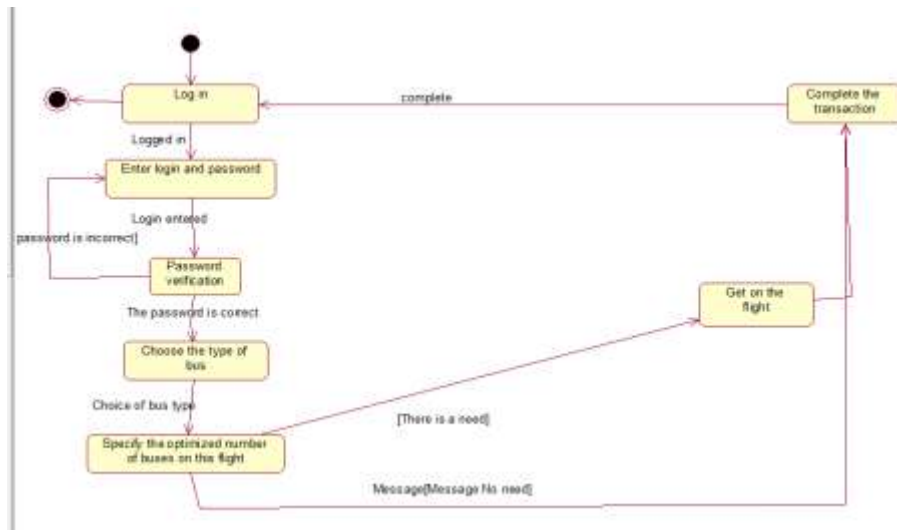


Fig. 4. State diagram

The deployment diagram shows the working copies of the program components, they are buses that belong to the transport company.

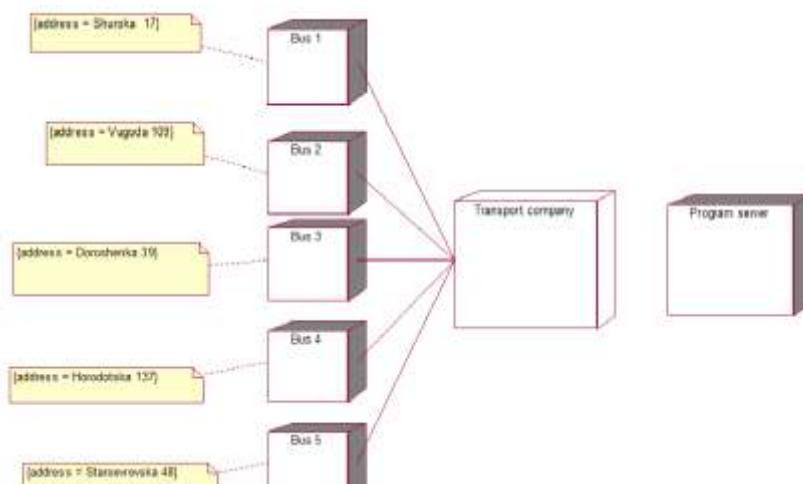


Fig. 5. Deployment diagram

3 Experiments

The next task was to calculate the main statistical characteristics of the process:

- minimum number of bus passengers;
- maximum number of bus passengers;
- the average value of the number of bus passengers;
- median;
- variance;
- standard deviation.

The code snippet allows you to calculate all of the above characteristics.

```
# basic statistical characteristics of the process
print("Min amount of people in bus:",min(dataset.inbus))
print("Max amount of people in bus:",max(dataset.inbus))
print("Mean:",np.mean(dataset.inbus))
print("Median:",np.median(dataset.inbus))
print("Variation:",np.var(list(dataset.inbus)))
print("Standart deviation:",np.std(dataset.inbus))
```

Fig. 6 shows the result of executing the above code snippet.

```
Min amount of people in bus: 0.0
Max amount of people in bus: 100.0
Mean: 59.90909090909091
Median: 72.5
Variation: 780.6280991735537
Standart deviation: 27.939722603733088
```

Fig. 6. The value of the main characteristics

The results obtained in Fig. 6 allow us to conclude that the minimum of passenger number in the bus = 0, the maximum is 100 people. The average number of passengers in the bus is 60 passengers, the median is 73 passengers, the variation = 780.63 and the standard deviation = 27.9..

Travel time along the entire route by one bus was determined. The distribution of time between stops was carried out in proportion to the length of the path between stops.

```
# time of passing the route by one bus
# time to overcome the route between stops by one bus
dataset['time'] = dataset['next_stop[km]']/40+2/60
time = np.ceil(sum(dataset.time)*60)
dataset['time_min'] = dataset['time']*60
print("Full time of bus on its way:",time)
col=['name', 'time_min']
print("Time interval between bus-stop for one bus:\n",
dataset[col])
```

Fig. 7 shows the full time of the entire route by one bus. Also, the time intervals of overcoming away by one bus in minutes between each stop according to the distance between them are tabulated.

Full time of bus on its way: 61.0
 Time interval between bus-stop for one bus:

	name	time_min
0	Zaliznychnyi vokzal	3.050
1	Prymiskyi vokzal	2.525
2	Skrynia Mall	2.675
3	Kulparkivska St	2.525
4	Kardiologichnyi tsentr	2.750
5	Antonovycha St	3.200
6	Kulparkiv	3.050
7	Sknylivok	2.750
8	Shchyretska St	2.900
9	Naukova St	2.525
10	Vasylia Symonenka St	2.525
11	Klub Naukovyi	2.525
12	Tsentr zainiatosti	2.525
13	Kniahyni Olhy St	2.600
14	Troleibusna St	3.275
15	Stryiska St - Naukova St	2.600
16	Skoryny St	2.600
17	Yaroslava Hasheka St	2.450
18	Mykhaila Maksymovycha St	2.675
19	Avtovokzal	2.750
20	Sokilnytska St	3.800
21	King Cross Leopold Mall	2.000

Fig. 7. Time intervals between bus-stops

For a predetermined λ and bus capacity, we calculate the number of buses on the route that would maximize the total number of passengers carried per flight.

The following code snippet performs the above task:

```
# optimal number of buses
# maximum number of transported passengers
count = 0
S= 0
while(True):#
    count+=1
    #1 зупинка
    dataset['comein']=random.randint(2,40)
    dataset['comeout'] = 0
    dataset['inbus'] = dataset['comein']
    dataset['queue'][0]=0
    for i in range(1,n): #зупинки
```

```

        da-
dataset['queue'][i]=np.ceil(abs(random.expovariate(alpha*(-
i)/n)))
        if(dataset.inbus[i-1]>70):
            dataset.comeout[i] = np.random.randint(0, da-
dataset.inbus[i-1]*0.8)
        elif(dataset.inbus[i-1]==0):
            dataset.comeout[i] = 0
        else:
            da-
ta-
set.comeout[i]=np.ceil(np.random.randint(0,dataset.inbus[
i-1]))
        dataset.inbus[i]=dataset.inbus[i-1]-
dataset.comeout[i]+dataset['queue'][i]
        if(dataset.inbus[i]>m):
            dataset.comein[i] = m-dataset.inbus[i-1]
            dataset['queue'][i] = dataset['queue'][i]-(m-
dataset.inbus[i-1])
            dataset.inbus[i] = m
        else:
            dataset['comein'][i]=dataset['queue'][i]
            dataset['queue'][i]=0
S+= dataset.comein.sum()
dataset['comeout'][n-1]=dataset['inbus'][n-2]
dataset['inbus'][n-1] = 0
f = 0
if(dataset[dataset['queue']>60].empty):
    break

```

Fig. 8 shows the number of buses that is optimal to maximize the transportation of people per flight and shows the number of people.

```

In [42]: count
Out[42]: 4

```

```

In [43]: 5
Out[43]: 1222

```

Fig. 8. Number of buses and passengers

The next step in working with this route will be to work on weekdays and weekends, as the number of passengers varies. That is, it was necessary to set the dependence of λ on the hour of the day for weekdays and weekends (including holidays). And calcu-

late the optimal number of buses on the route for working days and for weekends (holidays). Fragment of the program:

```
# number of passengers depending on the day of the week
and the time of day
# optimal number of buses
dataset_time = pd.DataFrame([[0.6,0.4],
                             [1,0.4],
                             [1,0.6],
                             [0.4,0.6],
                             [0.4,1],
                             [1,1],
                             [1,0.6],
                             [0.6,0.6]], columns = ['Work-
Day','Weekend'], index = ['6-8','8-10','10-12','12-
14','14-16','16-18','18-20','20-22'])
count_of_bus = pd.DataFrame(columns
=['WorkDay','Weekend'], index = ['6-8','8-10','10-
12','12-14','14-16','16-18','18-20','20-22'])
count_of_people = pd.DataFrame(columns
=['WorkDay','Weekend'], index = ['6-8','8-10','10-
12','12-14','14-16','16-18','18-20','20-22'])
for j in dataset_time.columns:
    for k in dataset_time.index:
        count = 0
        S=0
        while(True):
            count+=1
            #1 зупинка
            dataset['comein']=random.randint(2,40)
            dataset['comeout'] = 0
            dataset['inbus'] = dataset['comein']
            dataset['queue'][0]=0

            for i in range(1,n): #зупинки
                da-
ta-
set['queue'][i]=np.ceil(abs(random.expovariate((dataset_t
ime[j][k]-0.9)*(-i)/n)))
                if(dataset.inbus[i-1]>70):
                    dataset.comeout[i] =
np.random.randint(0, dataset.inbus[i-1]*0.8)
                elif(dataset.inbus[i-1]==0):
                    dataset.comeout[i] = 0
                else:
```



```

da-
ta-
set.comeout[i]=np.ceil(np.random.randint(0,dataset.inbus[
i-1]))
dataset.inbus[i]=dataset.inbus[i-1]-
dataset.comeout[i]+dataset['queue'][i]
if(dataset.inbus[i]>m):
dataset.comein[i] = m-
dataset.inbus[i-1]
dataset['queue'][i] = da-
taset['queue'][i]-(m-dataset.inbus[i-1])
dataset.inbus[i] = m
else:
da-
taset['comein'][i]=dataset['queue'][i]
dataset['queue'][i]=0
#останняя зупинка
S+= dataset.comein.sum()
dataset['comeout'][n-1]=dataset['inbus'][n-2]
dataset['inbus'][n-1] = 0
f = 0
if(dataset[dataset['queue']>60].empty):
break
count_of_bus[j][k]=count
count_of_people[j][k] = S
print(k,j)

```

Fig. 9 shows the optimal number of buses according to generated data, according to the time interval on weekdays and weekends (holidays).

Index	WorkDay	Weekend
6-8	1	1
8-10	11	1
10-12	2	1
12-14	1	1
14-16	1	9
16-18	7	2
18-20	3	2
20-22	2	2

Fig. 9. Number of buses

4 Discussion

Fig.10 shows the schedule of the distribution of preliminary data.

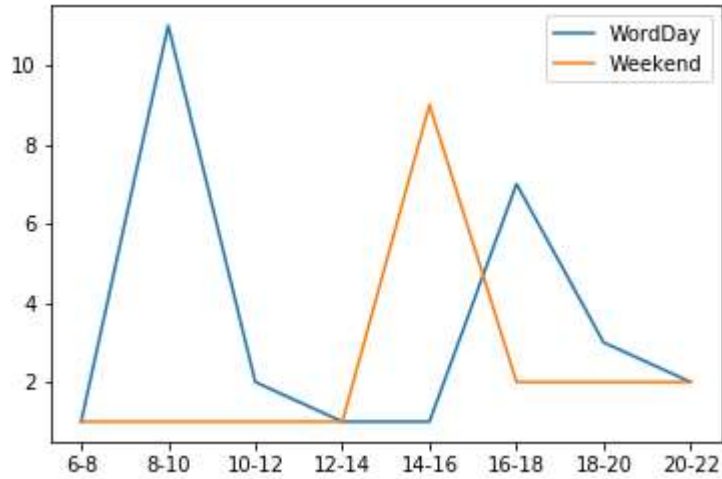


Fig. 10. Schedule of bus distribution according to the time of day

Fig.11 shows the approximate number of people according to my generated data, according to the time interval on weekdays and weekends (holidays) that use this route.

Index	WorkDay	Weekend
6-8	185	120
8-10	4701	144
10-12	921	244
12-14	131	280
14-16	152	3166
16-18	2837	912
18-20	1455	559
20-22	494	494

Fig. 11. Number of passengers who use the bus №10 in a certain period

Fig.12 shows the schedule of distribution of passenger flows on weekdays and week-ends of the week and according to the time of day.

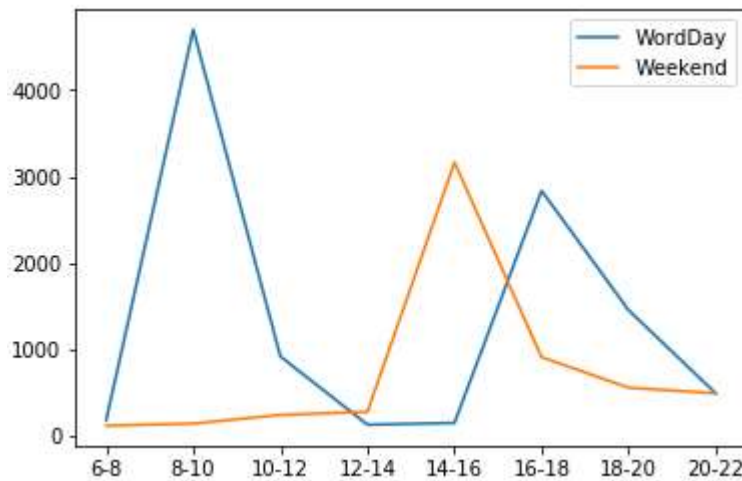


Fig. 12. Schedule of passenger traffic distribution

Finally, we find the estimate for the mathematical expectation and construct a confidence interval that will correspond to a confidence probability $\alpha = 0.8$ for all flights per day. Fragment of the program:

```
# score for mathematical expectation
# confidence interval for mathematical expectation
mat_spod = np.mean(count_of_people_in_run)
F= 1.282 #при значенні довірчої імовірності 0.8
D = np.var(count_of_people_in_run)
sigma = pow(D/len(count_of_people_in_run),1/2)
d1 = mat_spod - F * sigma
d2 = mat_spod + F * sigma
print("Math expectation:", mat_spod)
print("Confidence interval: (", d1,";", d2, ")")
```

Fig.13 shows the result of the execution of the above fragment of the program code for calculation of a point estimation of mathematical expectation and trust probability.

```
Math expectation: 3901.625
Confidence interval: ( 3845.286269849311 ; 3957.963730150689 )
```

Fig. 13. Evaluation of the expectations and confidence interval

Fig.14 shows the number of passengers who were transported by each of the 8 previously specified buses during the day.

```
count_of_people_in_run  
[3891, 3798, 4007, 3851, 3880, 3731, 4165, 3890]
```

Fig. 14. Number of passengers in each of the 8 buses

Fig.15 shows the graph of dependence of the total number of passengers on the number of buses. As we can see, with the increase of passengers, it is also necessary to increase the number of buses on the route for comfortable transportation of passengers without congestion of bases on the route.

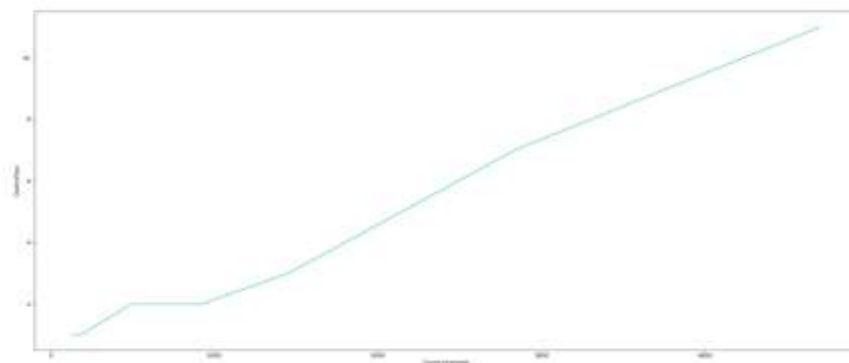


Fig. 15. Dependence Diagrams of the number of passengers on the number of buses

5 Conclusion

Because of work on optimizing the number of buses on the route, the theory of the queuing system is introduced. The main characteristics of QMS are considered. In addition, the end result is a program for analysis and processing of data on one of the routes of the city bus of the city of Lviv, namely: route number 10, which runs daily from six o'clock in the morning to ten o'clock in the evening every working day, and on weekends and holidays. Because of the analysis, the busload on the route is modeled, the main statistical characteristics of the process on route number 10 were calculated, and the optimal number of buses on the route is calculated, which would maximize the total number of passengers per flight. In particular, the developed program allows you to calculate the optimal number of buses on the route according to the previously specified criteria for specific hours of the day on weekdays (weekends) and weekends (including holidays). The work is illustrated by a sufficient number of schedules of passenger distribution, their dependence on the number of buses, schedule, etc.

References

1. System analysis. Theory of queuing, er.nau.edu.ua/bitstream/NAU/21000/23/Лекция_CA_12.pdf.
2. Theory of queuing, studopedia.com.ua/1_133235_teoriya-masovogo-obslugovuvannya.html.
3. Learn Data Science Online. DataCamp, www.datacamp.com.
4. Probability theory, eb.posibnyky.vntu.edu.ua/fitki/4tichinska_teoriya_jmovirnostej/65.htm.
5. Kuzmin, O., Bublyk, M.: Economic evaluation and government regulation of technogenic (man-made) damage in the national economy. In: Computer sciences and information technologies. Proceedings of the XIth International Scientific and Technical Conference (CSIT 2016), 37–39. (2016). DOI: <https://doi.org/10.1109/STC-CSIT.2016.7589863>
6. Matseliukh, Y., Vysotska, V., Bublyk, M.: Intelligent system of visual simulation of passenger flows. In: CEUR Workshop Proceedings, 2604, 906. (2020) <http://ceur-ws.org/Vol-2604/paper60.pdf>
7. Bublyk, M., Matseliukh, Y., Motorniuk, U., Terebukh, M.: Intelligent system of passenger transportation by autopiloted electric buses in Smart City. In: CEUR Workshop Proceedings, 2604, 1280. (2020) <http://ceur-ws.org/Vol-2604/paper81.pdf>
8. Maslak, O.I., Moroz, O.V., Moroz, M.M.: Specific features of city public transport financing (Kremenchuk case study). *Actual Problems of Economics* (2014)
9. Krislata, I., Katrenko, A., Lytvyn, V., Vysotska, V., Burov, Y.: Traffic flows system development for smart city. In: CEUR Workshop Proceedings, Vol-2565, 280–294. (2020)
10. Katrenko, A., Krislata, I., Veres, O., Oborska, O., Basyuk, T., Vasyliuk, A., Rishnyak, I., Demyanovskyi, N., Meh, O.: Development of Traffic Flows and Smart Parking System for Smart City. In: Computational Linguistics and Intelligent Systems, COLINS, CEUR workshop proceedings, Vol-2604, 730-745. (2020).
11. Peleshko D., Rak T., Lytvyn V., Vysotska V., Noennig J.: Drone monitoring system DROMOS of urban environmental dynamics. In: CEUR Workshop Proceedings, Vol-2565, 178–19. (2020)
12. Lytvyn, V., Kowalska-Styczen, A., Peleshko, D., Rak, T., Voloshyn, V., Noennig, J. R., Vysotska, V., Nykolyshyn, L., Pryshchepa, H.: Aviation Aircraft Planning System Project Development. In: *Advances in Intelligent Systems and Computing IV*, Springer, Cham, 1080, 315-348. (2020)
13. Lytvyn, V., Dmytriv, A., Berko, A., Aliksieiev, V., Basyuk, T., Noennig, J., Peleshko, D., Rak, T., Voloshyn, V.: Conceptual Model of Information System for Drone Monitoring of Trees' Condition. In: Computational Linguistics and Intelligent Systems, COLINS, CEUR workshop proceedings, Vol-2604, 695-714. (2020).
14. Aliksieiev, V., Markovych, B.: Implementation of UAV for environment monitoring of a Smart City with an airspace regulation by AIXM-format data streaming. In: *Industry 4.0* – Scientific-Technical Union of Mechanical Engineering “Industry 4.0”, Sofia, Bulgaria, 5(2/2020), 90–93. (2020), <https://stumejournals.com/journals/i4/2020/2/90>
15. Kunanets, N., Matsiuk, H.: Use of the Smart City Ontology for Relevant Information Retrieval. In: CEUR Workshop Proceedings, Vol-2362, 322-333. (2019)
16. Batiuk, T., Vysotska, V., Lytvyn, V.: Intelligent System for Socialization by Personal Interests on the Basis of SEO-Technologies and Methods of Machine Learning. In: CEUR workshop proceedings, Vol-2604, 1237-1250. (2020).
17. Kravets, P., Lytvyn, V., Vysotska, V., Burov, Y.: Promoting training of multi-agent systems. In: CEUR Workshop Proceedings, Vol-2608, 364-378. (2020)

18. Lytvyn, V., Vysotska, V., Osypov, M., Slyusarchuk, O., Slyusarchuk, Y.: Development of intellectual system for data de-duplication and distribution in cloud storage. In: *Webology*, 16(2), pp. 1-42. (2019)
19. Vysotsky, A., Lytvyn, V., Vysotska, V., Dosyn, D., Lyudkevych, I., Antonyuk, N., Naum, O., Vysotskyi, A., Chyrun, L., Slyusarchuk, O.: Online Tourism System for Proposals Formation to User Based on Data Integration from Various Sources. In: *Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT*, 92-97. (2019)
20. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: The Risk Management Modelling in Multi Project Environment.. In: *Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT*, 32-35. (2017)
21. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs. In: *Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT*, 113-116. (2018)
22. Lytvyn, V., Vysotska, V., Burov, Y., Demchuk, A.: Architectural ontology designed for intellectual analysis of e-tourism resources. In: *Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT*, 335-338. (2018)
23. Antonyuk, N., Vysotsky, A., Vysotska, V., Lytvyn, V., Burov, Y., Demchuk, A., Lyudkevych, I., Chyrun, L., Chyrun, S., Bobyk, I.: Consolidated Information Web Resource for Online Tourism Based on Data Integration and Geolocation. In: *Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT*, 15-20. (2019)
24. Artemenko, O., Pasichnyk, V., Kunanets, N., Shunevych, K.: Using sentiment text analysis of user reviews in social media for e-tourism mobile recommender systems. In: *Computational Linguistics and Intelligent Systems, COLINS, CEUR workshop proceedings, Vol-2604*, 259-271. (2020).
25. Shakhovska, N., Shakhovska, K., Fedushko, S.: Some Aspects of the Method for Tourist Route Creation. In: *Advances in Artificial Systems for Medicine and Education II*, 902, 527-537. (2019)
26. Antonyuk, N., Medykovskyy, M., Chyrun, L., Dverii, M., Oborska, O., Krylyshyn, M., Vysotsky, A., Tsiura, N., Naum, O.: Online Tourism System Development for Searching and Planning Trips with User's Requirements. In: *Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020*, 1080, 831-863. (2020)
27. Lozynska, O., Savchuk, V., Pasichnyk, V.: Individual Sign Translator Component of Tourist Information System. In: *Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, Springer, Cham*, 1080, 593-601. (2020)
28. Savchuk, V., Lozynska, O., Pasichnyk, V.: Architecture of the Subsystem of the Tourist Profile Formation. In: *Advances in Intelligent Systems and Computing*, 871, 561-570. (2019)
29. Stoyanova-Doycheva, A., Ivanova, V., Glushkova, T., Stoyanov, S., & Radeva, I.: dynamic generation of cultural routes in a tourist guide. *International Journal of Computing*, 19(1), 39-48. (2020). <http://computingonline.net/computing/article/view/1691>
30. Berko, A., Alieksieiev, V.: A Method to Solve Uncertainty Problem for Big Data Sources. In: *International Conference on Data Stream Mining and Processing, DSMP*, 32-37. (2018)
31. Berko, A.Y., Aliekseyeva, K.A.: Quality evaluation of information resources in web-projects. In: *Actual Problems of Economics*, 136(10), 226-234. (2012)
32. Berko, A.Y.: Models of data integration in open information systems. In: *Actual Problems of Economics*, (10), 147-152. (2010)