POZNAN UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTING AND TELECOMMUNICATIONS

Institute of Radiocommunications

Simulation techniques

final report

Joanna Szewczyk

132418

Teleinformatics, IIst

POZNAŃ 2021

Table of Contents

1. TASK TO BE SOLVED DURING THE PROJECT

Simulate a transport company, where $\boldsymbol{k}$ trucks with capacity of $\boldsymbol{u}$ units transport cargo goods between the headquarters and $\boldsymbol{n}$ regional depots. Goods arrive in batches of size $\boldsymbol{r}$ units, where $\boldsymbol{r}$ is a random variable following normal distribution with mean $\boldsymbol{\mu_r}$ and variance $\boldsymbol{\sigma_r^2}$ (but the lowest possible size is 0.1 unit and the highest is 10 units). In the headquarters the cargo batch destination depot $\boldsymbol{d_n}$ is a random variable following uniform distribution with equal probability for each depot, while the cargo batch destination for all cargo generated in depots is the headquarters. The time interval between the arrival (generation) of two consecutive cargo batches is a random variable with exponential distribution and average $\boldsymbol{c_{hq}}$ and $\boldsymbol{c_d}$ for the headquarters and each depot, respectively. Cargo batches wait in queues at the headquarters and the depots until they are taken by the next truck going to the destination point of a cargo batch, where the truck destination at the headquarters is selected with strategy **S**, while for the depots the destination is always the headquarters. After the destination is determined the cargo batches destined to this point are loaded in first-in first-out order (FIFO) until the next one exceeds the truck capacity. Only full batch can be loaded, assuming there is enough space in the truck. In the headquarters there are $\boldsymbol{M}$ loading/unloading platforms, while in each depot there are $\boldsymbol{N}$ loading/unloading platforms, with single platform able to accommodate one truck. Trucks upon arrival enter a free platform unless all of them are occupied – in such a case they wait in queue. Then they are unloaded and later loaded, with the loading and unloading time for each cargo batch determined as $\boldsymbol{r*T_l}$ and $\boldsymbol{r*T_u}$, respectively. Delivered cargo batches leave the system. Trucks leave the platform according to strategy $\boldsymbol{P}$. The traveling time for each truck between any depot and the headquarters follows normal distribution with average $\boldsymbol{\mu_t}$ and variance $\boldsymbol{\sigma_t^2}$.

Develop a C++ project that simulates the transport company described above and determine:

- the average queuing time for a cargo batch,
- the average number of queued cargo units in headquarters and depots, respectively,
- the average transport time for a cargo batch (time from arrival in the system until they are delivered – unloaded),
- the average utilization of trucks (time of service – loading/unloading or traveling - to the total time).

Parameters: MT4 + S2 + P1 +D1

	D1
k	8
u	10
n	6
(μ_r, σ_r^2)	(2.0, 0.6^2)
c_{hq}	2.2
c_d	12.3
(M, N)	(3, 1)
T_l	0.5
T_u	0.7
(μ_t, σ_t^2)	(20.0, 1.6^2)

Destination selection strategy S:

S2 - Destination depot is selected randomly according to uniform distribution, with equal probability of each depot.

Trucks departure strategy P:

P1 - Trucks leave headquarters/depot only when the next cargo batch cannot be loaded (exceeds truck capacity).

Simulation method MT:

MT4 - Process interaction.

2. DESCRIPTION OF THE SIMULATION MODEL

The simulated company model consists of 8 trucks with capacity of 10 units, goods, one headquartes and 6 regional depots.

2.1. Diagram

Below is a diagram showing the structure of the implemented simulation model (Fig. 2.1).

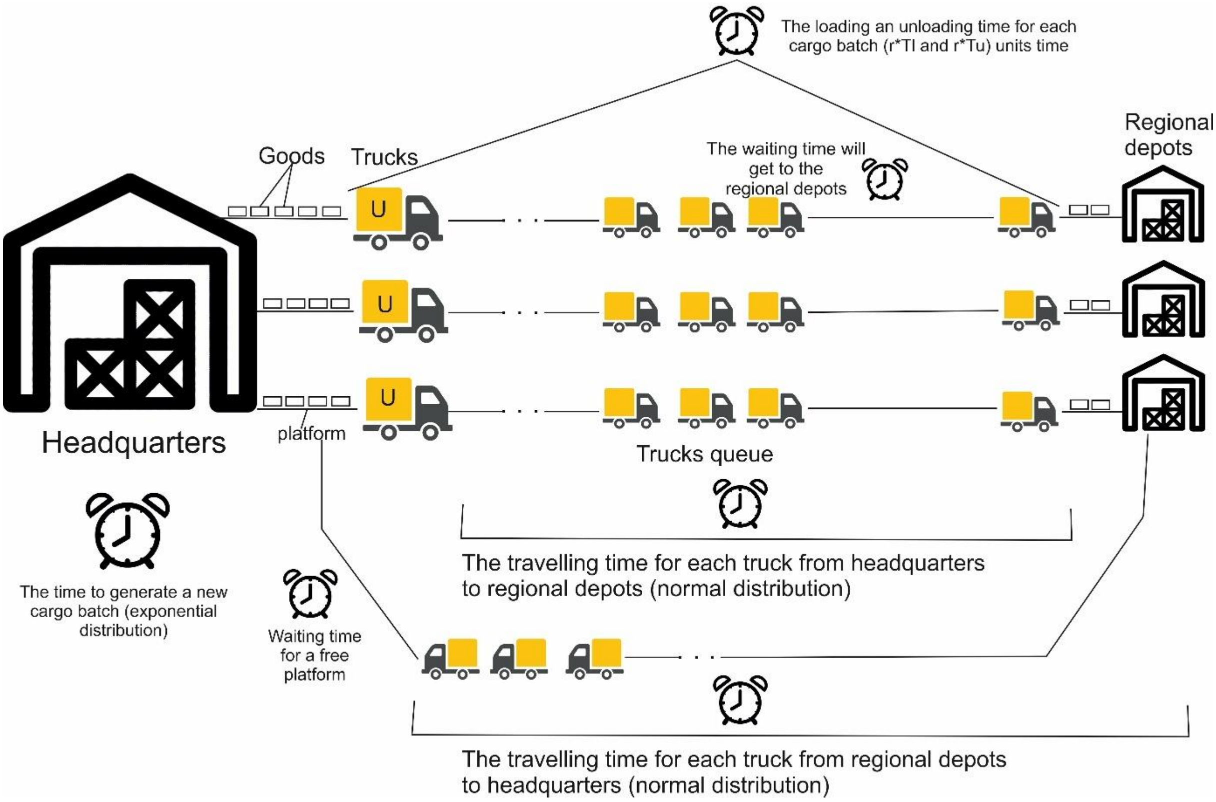


Fig. 2.1. Simulation model.

2.2. Implemented object

The implemented simulation model includes the objects presented in the table below.

Object	Class name/struct	Description	Attribute
Transport company	TransportCompany	Class used as a container of all other objects.	- pointer on all regional depots, - pointer on platform in HQ, - id goods, - variable of average time truck, - variable of average counter - function to collecting data about average Time Pack in queue, - functions to Print Statistic, - Function to save data to file,
Platform	Platform	Class that represents a Headquarters.	- information about the number of the platform in HQ, - queue and vector to pack process, -functions supporting the parcel process and everything related to parcels letting and letting trucks on the platform as well as loading and unloading.

Object	Class name/struct	Description	Attribute
Truck	Truck	Class that represents a single truck.	- Pointer on transport company - Execute functions
Goods	Goods	Class that represents a single goods.	- id, - size, - Pointer on transport company - execute functions
Regional depots	RegionalDepots	Class that represents a single regional depots.	- information about the number of the platform in all RD, - queue and vector to pack process, - size platform -functions supporting the parcel process and everything related to parcels letting and letting trucks on the platform as well as loading and unloading.
Process	Process	Class responsible for the operation of processes in the program	- Process phase, -id_ - truck id and random package id -idp - unique id of consecutive packages, for the truck it is 0 -idx_ - identifier where the truck is to go, for package 0

Object	Class name/struct	Description	Attribute
			- size, for a 10 truck, for a package randomly selected according to the schedule - Vector on pack list - Pointer on event list
Logger	Logger	Class responsible for defining the amount of displayed details	Functions required for varying degrees of detail display
Event List	Event_list	class to handle events	- function responsible for adding events - pointer on first and last event
Generators	Generators	A class that handles all generators	- Seed - Function of normal generator - Function of Exponential generator - Finction of uniformly generator.

3. DESCRIPTION OF PROCESSES AND BLOCK DIAGRAMS

3.1. Processes and block diagrams for object Truck

3.1.1. Processes for Truck

Phase 1:

1. drive to the appropriate receiver of goods,
2. waiting in queue for a free platform,
3. if platfotm is free, go to phase 2,

Phase 2:

4. entering truck on the first free platform,
5. make platform busy and start service,
6. unloading goods, and go to phase 3.

Phase 3:

1. loading goods, and go to phase 4.

Phase 4:

1. finish loading,
2. check state truck,
3. if truck isn’t full, wait for a new delivery of goods, and back to phase 3,
4. if truck is full, go to the phase 5.

Phase 5:

1. if truck is in HQ, take the ID of the package specifying the depot,
2. if truck is in depot, drive to HQ,
3. go to phase 1.

3.1.2. Block diagrams for Truck

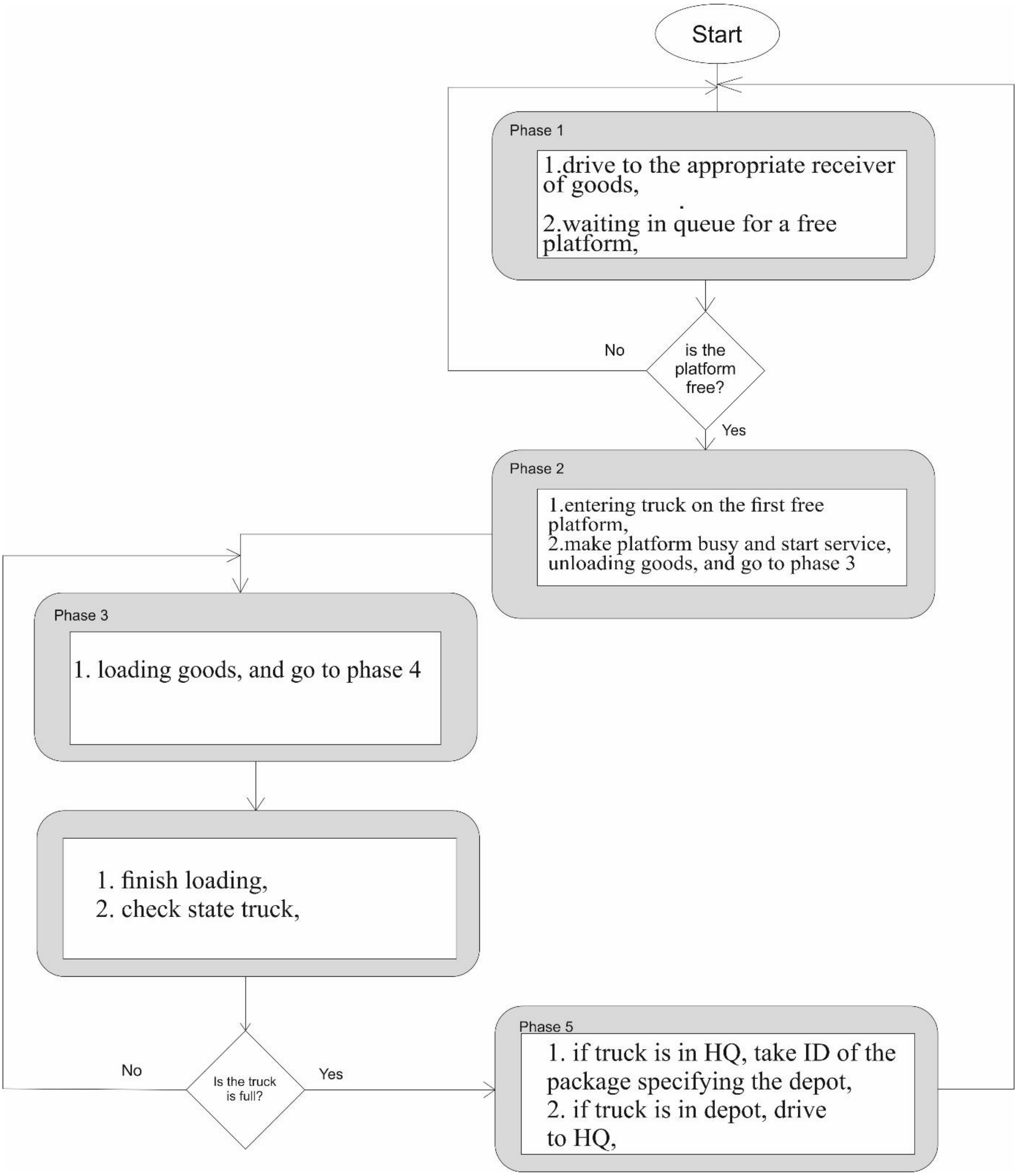


Fig. 3.1. Block diagram of truck life cycle.

3.2. Processes and block diagrams for loading/unloading service

3.2.1. Processes for loading/unloading service

Phase 1:

1. start a generate goods,
2. place goods in line to platform,
3. generate driving ID in goods,
4. if platform is free, go to next phase.

Phase 2:

1. allow the truck to enter the platform,
2. start unloading truck,
3. send goods from line to platform, and go to phase 3.

Phase 3:

1. start loading truck,
2. if truck is full, stop service, and go to next phase.

Phase4:

1. check the queue of waiting trucks,
2. if platform is free go to phase 1.

3.2.2. Block diagram for loading/unloading service

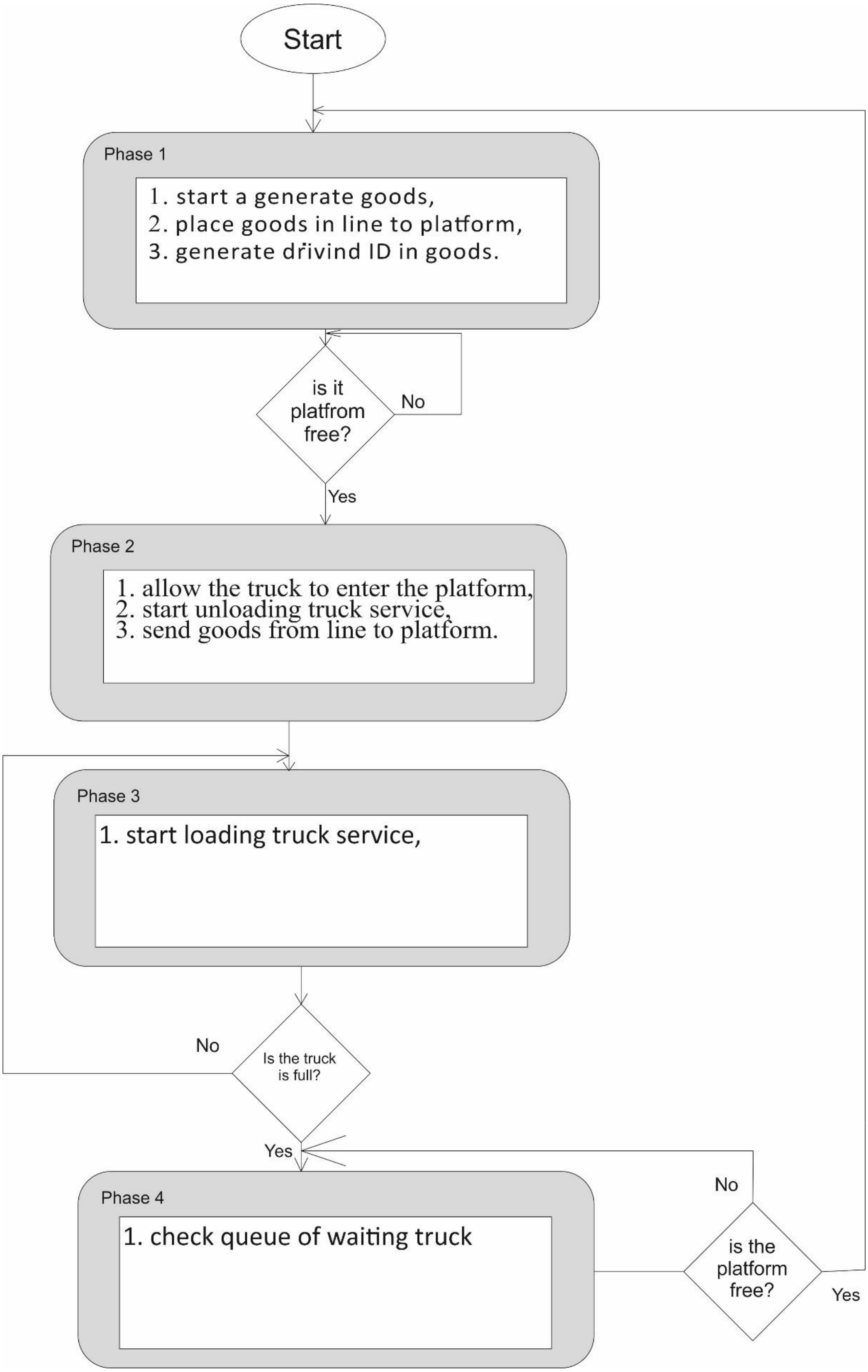


Fig. 3.2. Block diagram of loading/unloading service life cycle.

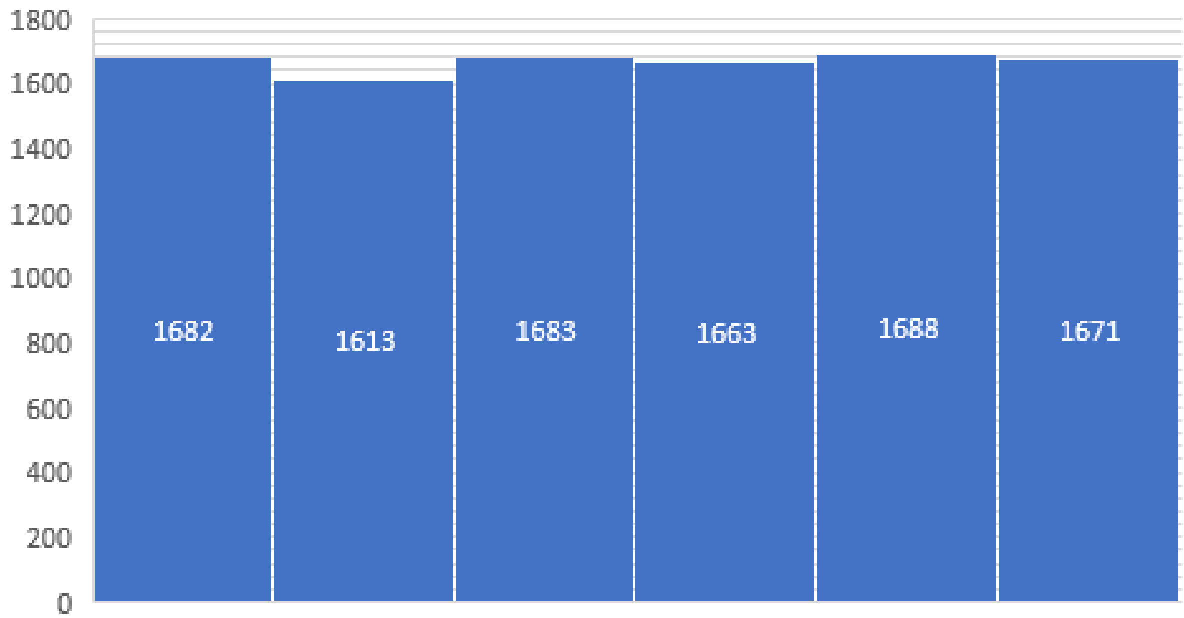
4. IMPLEMENTATION

4.1. Generators

Below are the histograms of 10,000 samples made from the generators. They were made to check the correct operation of the generators.

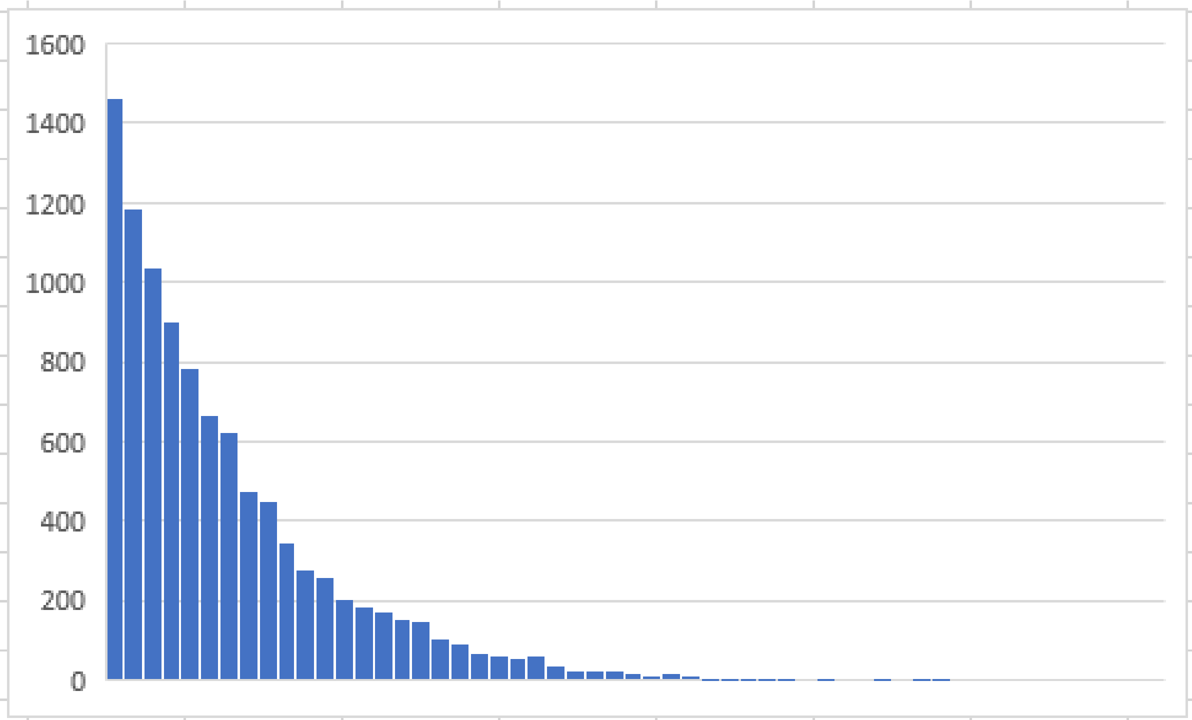
4.1.1. uniformly distributed numbers

```
int Generators::Distribution()
{
    static::default_random_engine generator(kSeed);
    uniform_int_distribution<int>distribution(1, 6);
    return distribution(generator);
}
```



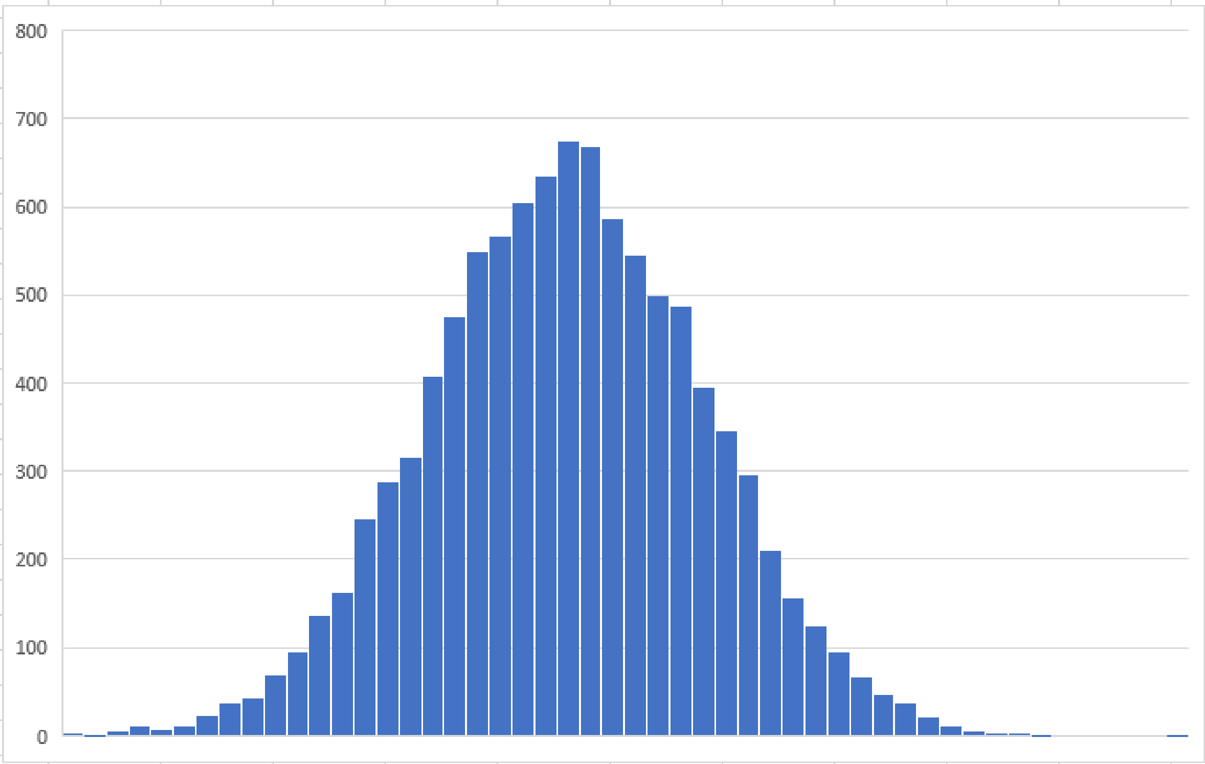
4.1.2. exponential distribution

```
double Generators::ExponentialDistributionGenerator(double average)
{
    static default_random_engine generator(kSeed);
    const exponential_distribution<double> distribution(1/average);
    return distribution(generator);
}
```



4.1.3. normal distribution

```
double Generators::NormalDistributionGenerator(const pair<const double, const double> p)
{
    static default_random_engine generator(kSeed);
    normal_distribution<double> distribution(p.first, p.second);
    return distribution(generator);
}
```




```
ofstream f;
double tim;
for (int i = 0; i < 10000; i++)
{
    //tim = Generators::Distribution();
    //tim = Generators::ExponentialDistributionGenerator(12);
    tim = Generators::NormalDistributionGenerator(make_pair(2.00, 0.36));

    f.open("excel/NormalDistributionGenerator.csv", ios::app);
    //f.open("excel/Distribution.csv", ios::app);
    //f.open("excel/ExponentialDistributionGenerator.csv", ios::app);
    f << std::to_string(tim) << endl;
    f.close();
}
//Generators::Distribution();
//Generators::NormalDistributionGenerator(make_pair(2.00, 0.36));
```

The above code shows the generation of 10000 samples for all generators.

4.2. Input parameters

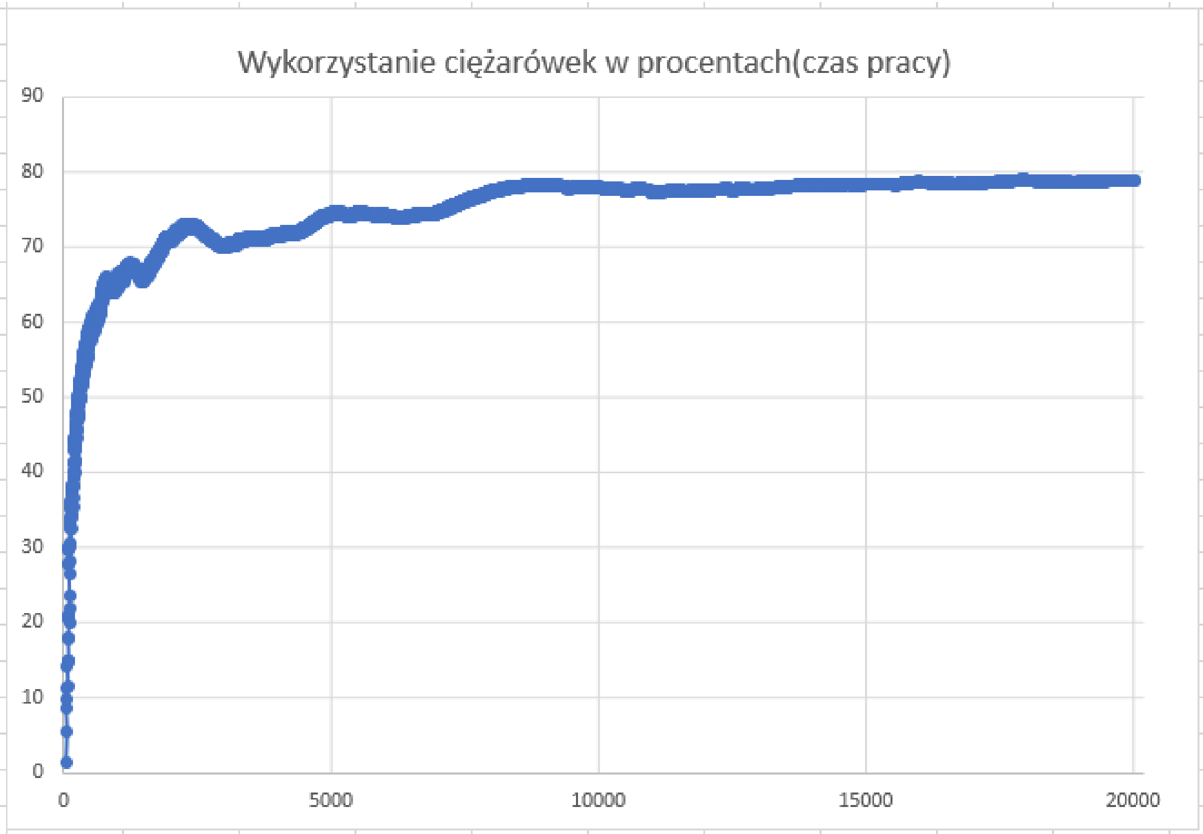
Number of Truck 8	**Simulation time** 5000ms	**Number of simulations** 10
Number Of RD 6	**Number of platform in HQ** 3	
Inicjal phase 1000ms	**Start seed** 100000	

A large part of the parameters of the implemented digital simulation was given at the beginning of the task, including: number of trucks = 8, number of depots = 6, or the number of platforms in individual warehouses. The remaining parameters, not imposed in advance, had to be determined while creating the program. The optimal simulation time of 5000 ms was selected to test the simulation and determine the initial phase. The initial grain value was set at 100,000 and was increased with each subsequent measurement to check the randomness of the simulation.

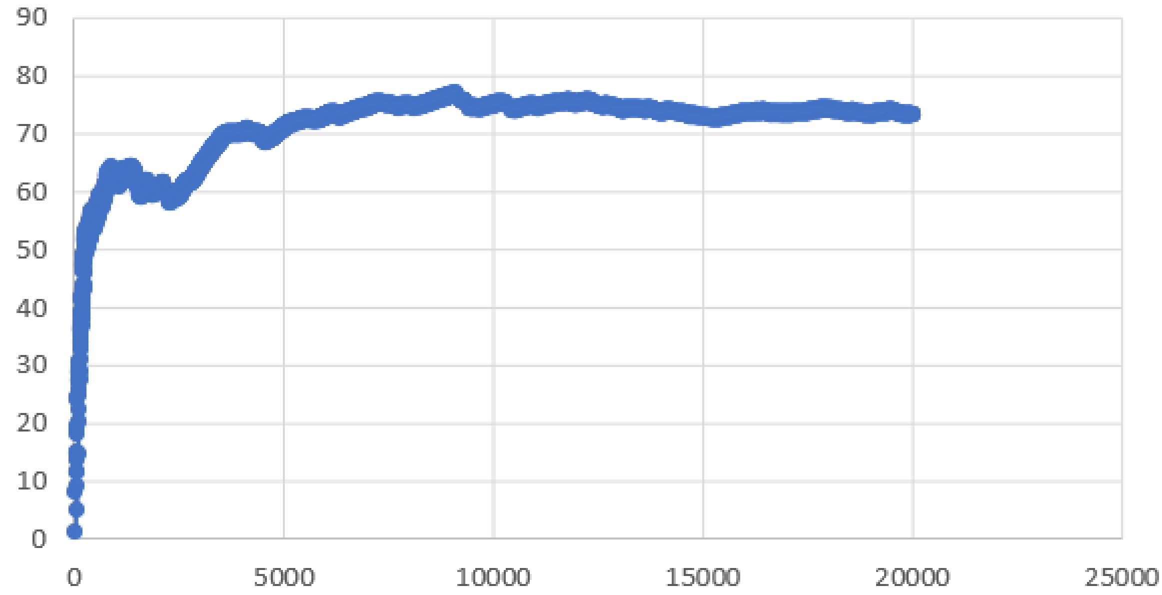
4.3. Initial phase determination

The initial phase ends at about 1000ms. This is the moment when the chart begins to stabilize.

The average simulation results are observed in the program console. A chart has also been created showing the percentage of average truck utilization. Thanks to the graph, the initial phase of the simulation can be determined.



Graph for the ID assigned based on the first batch removed from the queue



Graph based on the ID generated from an evenly distributed generator

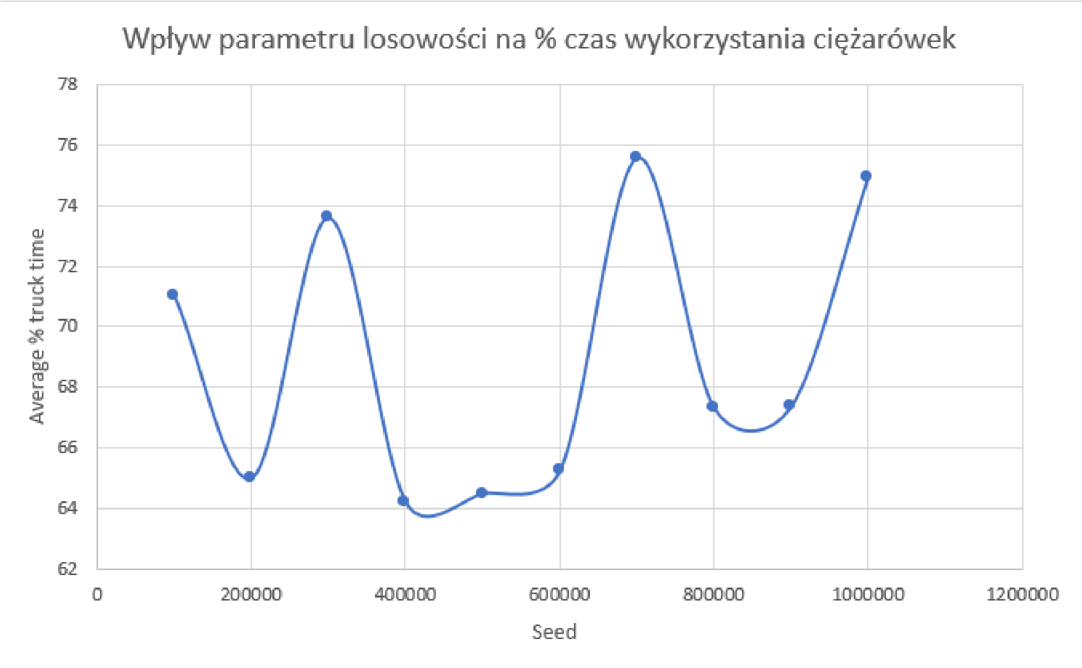
```
**************************Statistic*********************************************
Avarage percentage truck time use: 79.6281

Avarage truck time use: 61.0688

*****************************HQ*************************************************
Avarage pack queue size: 57
Avarage time pack in queue: 131.906
*****************************RD*************************************************
RD1
Avarage pack queue size: 29
Avarage time pack in queue: 363.872
RD2
Avarage pack queue size: 135
Avarage time pack in queue: 1573.38
RD3
Avarage pack queue size: 13
Avarage time pack in queue: 157.17
RD4
Avarage pack queue size: 57
Avarage time pack in queue: 653.82
RD5
Avarage pack queue size: 70
Avarage time pack in queue: 832.298
RD6
Avarage pack queue size: 41
Avarage time pack in queue: 529.125
```

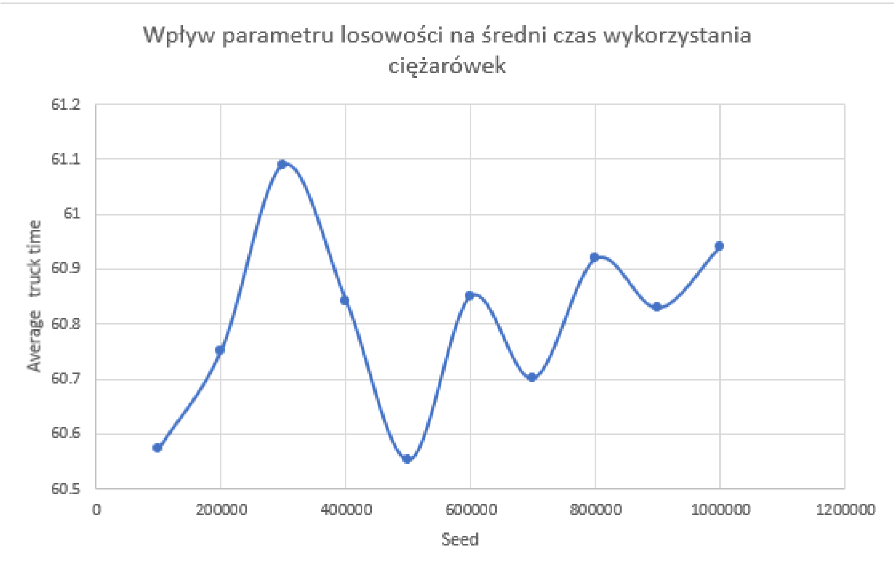
5. RESULTS

Lp.	kSeed	Average % truck time use	Average truck time use [ms]	Average pack queue size in HQ	Average time pack in queue in HQ [ms]	Average pack queue size in RD	Average time pack in queue in RD [ms]
1	100000	71.0033	60.5734	176	399.294	RD1: 33 RD2: 48 RD3: 50 RD4: 77 RD5: 11 RD6: 30	RD1: 370.39 RD2: 567.324 RD3: 678.384 RD4: 1071.26 RD5: 132.634 RD6: 388.045
2	200000	65.0037	60.7526	207	430.293	RD1: 56 RD2: 61 RD3: 75 RD4: 6 RD5: 55 RD6: 35	RD1: 625.645 RD2: 709.765 RD3: 905.356 RD4: 60.9976 RD5: 633.809 RD6: 442.4
3	300000	73.6095	61.0906	252	571.566	RD1: 42 RD2: 41 RD3: 53 RD4: 99 RD5: 57 RD6: 8	RD1: 529.081 RD2: 547.944 RD3: 679.424 RD4: 1475.29 RD5: 754.406 RD6: 91.6781
4	400000	64.1813	60.8426	240	511.622	RD1: 73 RD2: 59 RD3: 43 RD4: 28	RD1: 806.894 RD2: 739.775 RD3: 604.37 RD4: 342.033

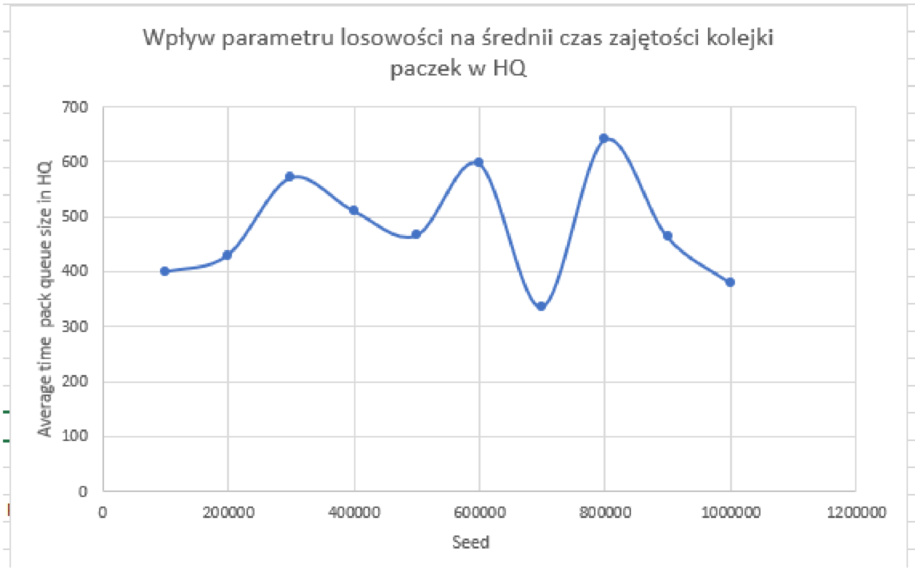
Lp.	kSeed	Average % truck time use	Average truck time use [ms]	Average pack queue size in HQ	Average time pack in queue in HQ [ms]	Average pack queue size in RD	Average time pack in queue in RD [ms]
						RD5: 23 RD6: 101	RD5: 286.686 RD6: 1271.08
5	500000	64.4867	60.5531	213	465.917	RD1: 79 RD2: 37 RD3: 51 RD4: 19 RD5: 51 RD6: 68	RD1: 974.646 RD2: 469.617 RD3: 578.365 RD4: 203.966 RD5: 586.716 RD6: 893.161
6	600000	65.2518	60.8513	270	597.543	RD1: 17 RD2: 69 RD3: 31 RD4: 58 RD5: 26 RD6: 82	RD1: 222.425 RD2: 916.794 RD3: 868.541 RD4: 814.515 RD5: 317.322 RD6: 1004.6
7	700000	75.594	60.7013	150	335.469	RD1: 43 RD2: 58 RD3: 89 RD4: 52 RD5: 23 RD6: 50	RD1: 449.028 RD2: 761.5 RD3: 1034.5 RD4: 695.552 RD5: 254.462 RD6: 613.193
8	800000	67.312	60.9194	291	640.948	RD1: 111 RD2: 9 RD3: 62 RD4: 55	RD1: 1509.77 RD2: 124.192 RD3: 767.101 RD4: 738.898

Lp.	kSeed	Average % truck time use	Average truck time use [ms]	Average pack queue size in HQ	Average time pack in queue in HQ [ms]	Average pack queue size in RD	Average time pack in queue in RD [ms]
						RD5: 21 RD6: 19	RD5: 304.878 RD6: 468.79
9	900000	67.3617	60.8291	201	463.321	RD1: 63 RD2: 76 RD3: 21 RD4: 67 RD5: 27 RD6: 85	RD1: 934.646 RD2: 806.86 RD3: 245.371 RD4: 837.777 RD5: 372.082 RD6: 1183.88
10	1000000	74.9525	60.9417	173	378.722	RD1: 44 RD2: 41 RD3: 35 RD4: 13 RD5: 38 RD6: 55	RD1: 504.094 RD2: 536.216 RD3: 479.525 RD4: 160.902 RD5: 404.937 RD6: 698.365
Average		68.87565	60.80551	217.3	479.4695	48.483	624.198
Confidence interval		68.50666-69.24464	60.79105-60.819966	215.2-219.4	476.3927-482.5463	46.087-50.879	615.411-632.985


Wpływ parametru losowości na % czas wykorzystania ciężarówek
Average % truck time
78
76
74
72
70
68
66
64
62
0
200000
400000
600000
800000
1000000
1200000
Seed


Wpływ parametru losowości na średni czas wykorzystania ciężarówek
Average truck time
61.2
61.1
61
60.9
60.8
60.7
60.6
60.5
0
200000
400000
600000
800000
1000000
1200000
Seed


Wpływ parametru losowości na średnią zajętość kolejki paczek w HQ
Average pack queue size in HQ
350
300
250
200
150
100
50
0
0
200000
400000
600000
800000
1000000
1200000
Seed



Wpływ parametru losowości na średnii czas zajętości kolejki paczek w HQ
Average time pack queue size in HQ
0
100
200
300
400
500
600
700
0
200000
400000
600000
800000
1000000
1200000
Seed

6. Conclusion

Simulation works as desired. While analyzing the obtained results, no obvious irregularities were noticed, which meant that the program was not working properly. In order to check it I analysed multiple times how program works. After many trials and tests, it can be stated that the created system works correctly and it complies with the process interaction method. I also created additional simple loop in order to examine the behaviour of random generators. Gathered data seems to be good – average times and size are probable.