ECCO XXXVI. Conference
Chania, Crete, Greece
11th May 2023.

Design and analysis of vehicle scheduling and routing methods on a port logistics problem from the aspect of environmental impact and cost-efficiency

Tibor Dulai, Daniil Baldouski, Balázs Dávid, György Dósa, Miklós Krész, Zsuzsanna Nagy, Ágnes Werner-Stark

Tibor Dulai acknowledges the financial support of the National Research, Development and Innovation Office of Hungary through the grant MEC_R 141075. Balázs Dávid and Miklós Krész gratefully acknowledge the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program, and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European Regional Development Fund). They are also grateful for the support of the Slovenian Research Agency (ARRS) through the grant BI-HU/21-22-010.

Agenda

- Presentation of the problem
- Basic structure of the port
- The customizable parameters of the port structure
- The framework – how the different components of the port are handled
- Input data structure
- Input classes
- Algorithm variants
- Analysis of the results

The problem

We intend to create an event-based simultaion **framework** for a port logistics system, **routing trucks and scheduling trucks and ships**.

The **port structure** should be customized in the framework.

We intend to generate **different classes of input data**.

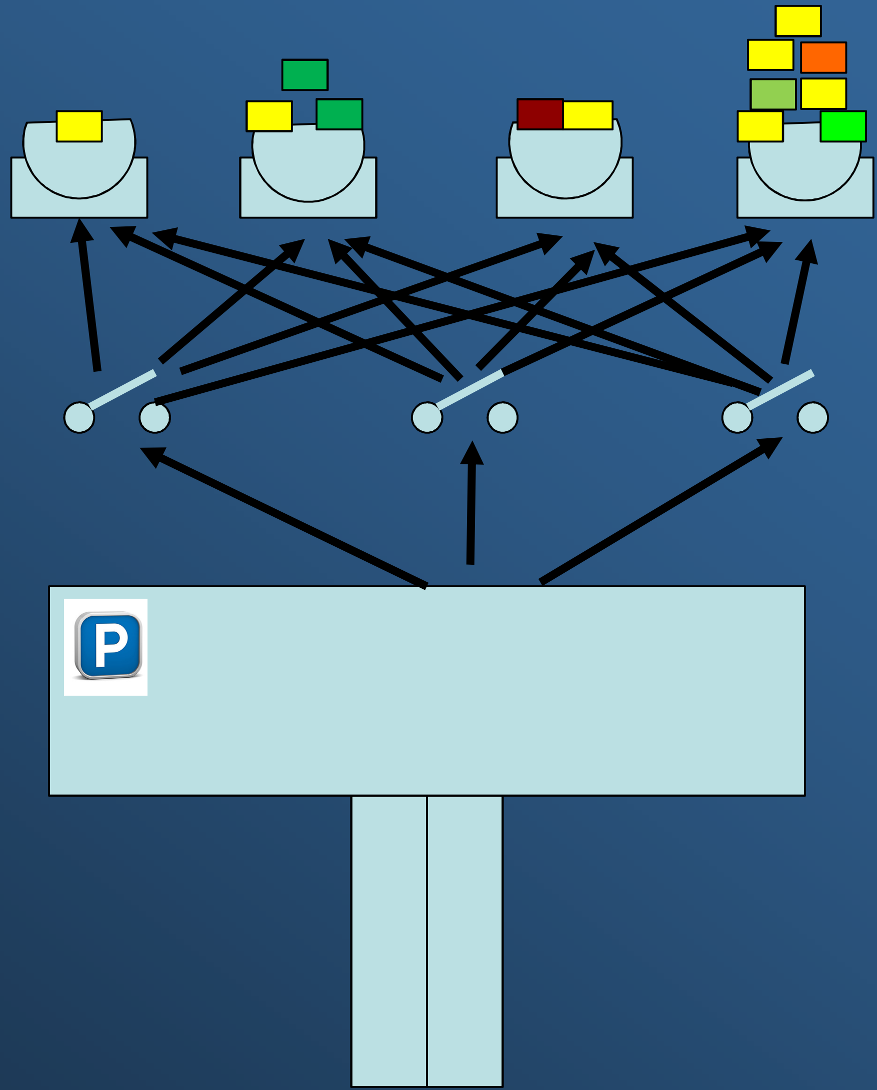
Some **decision points** of the system should be determined.

Our additional goal is to develop **algorithms** with different behaviour for the decision points.

We intend to **analyze** the efficiency of the algorithms on the input classes.

The structure of the port

Component's name	Component's capacity for trucks
Docks	1
	$c_{i,j}$
Gates	1
	c_i
Parking lot	C
Road	∞



The customizable parameters of the port structure

- number of gates : **G**
- number of docks : **D**
- type of the i^th gate: $type_i$, $i \in [1, \ldots, G]$
- capacity of the parking lot: **C**
- capacity of the j^th dock for containers: $c^{container}_j$, $j \in [1, \ldots, D]$
- time to unload a container in the j^th dock: t^{unload}_j, $j \in [1, \ldots, D]$
- time to load a container into a ship in the j^th dock: t^{load}_j, $j \in [1, \ldots, D]$
- time from the parking lot to the i^th gate: t_i, $i \in [1, \ldots, G]$
- time from the i^th gate to the j^th dock: $t_{i,j}$, $i \in [1, \ldots, G], j \in [1, \ldots, D]$
- capacity of the route from the parking lot to the i^th gate: c_i, $i \in [1, \ldots, G]$
- capacity of the route from the i^th gate to the j^th dock: $c_{i,j}$, $i \in [1, \ldots, G]$, $j \in [1, \ldots, D]$

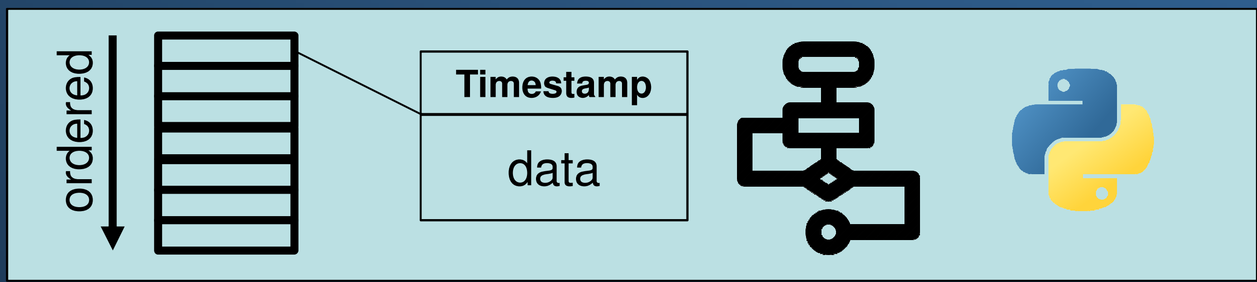
The simulation framework

- written in Python language
- event-driven approach: simulated timer-based event handling

Input

Port structure /json/ | Input data (trucks/ships) /json/

Framework



Output

Proposed schedule

Operation of the simulation framework
The Road (before the parking lot)

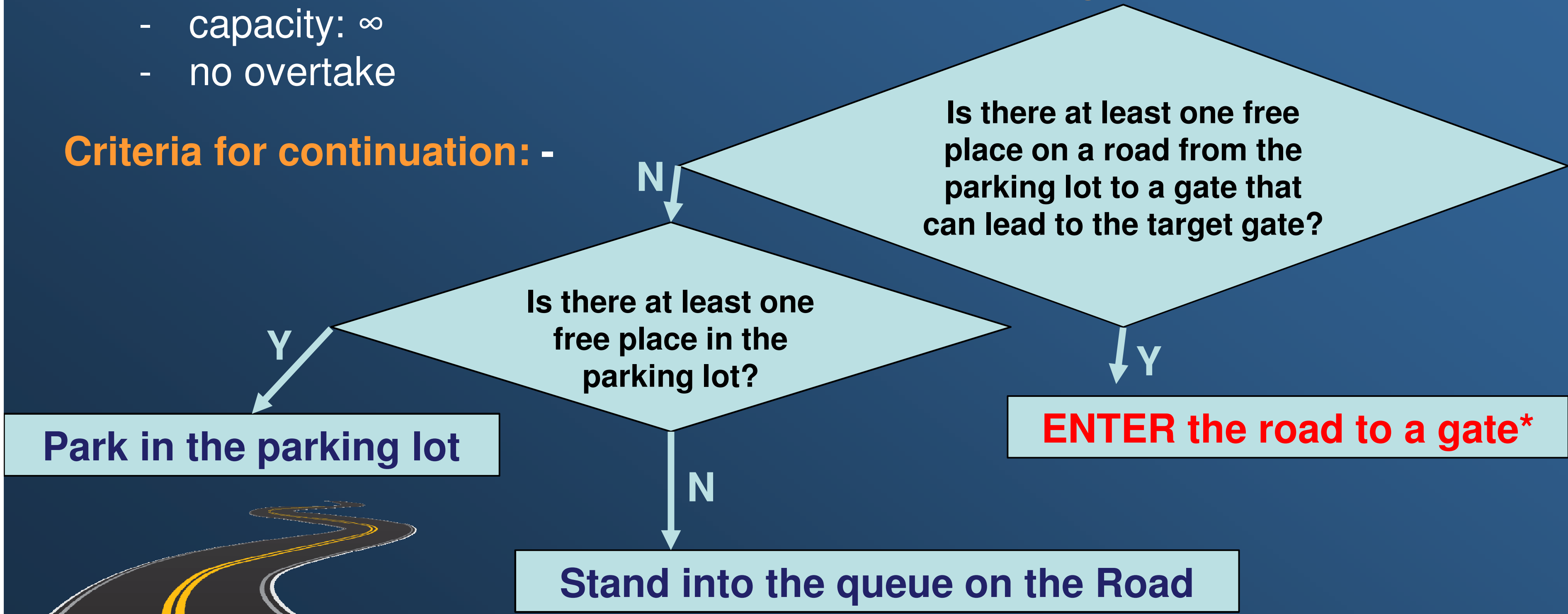
Constraints:

- works in FIFO manner
- capacity: ∞
- no overtake

Criteria for continuation: -

Event: Arrival of a truck

Event handling:



Operation of the simulation framework
The Road (before the parking lot)

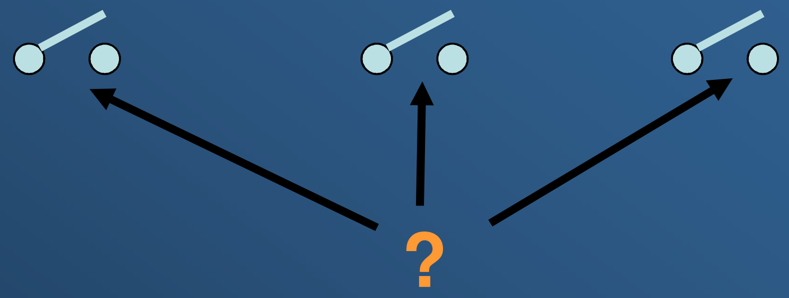
ENTER the road to a gate*

Decision point: to which Gate to direct the truck?

Applied algorithms:

1. Choose the gate that results in **minimal overall cost** for the truck

2. Choose an appropriate gate **randomly**



Operation of the simulation framework
The Gate

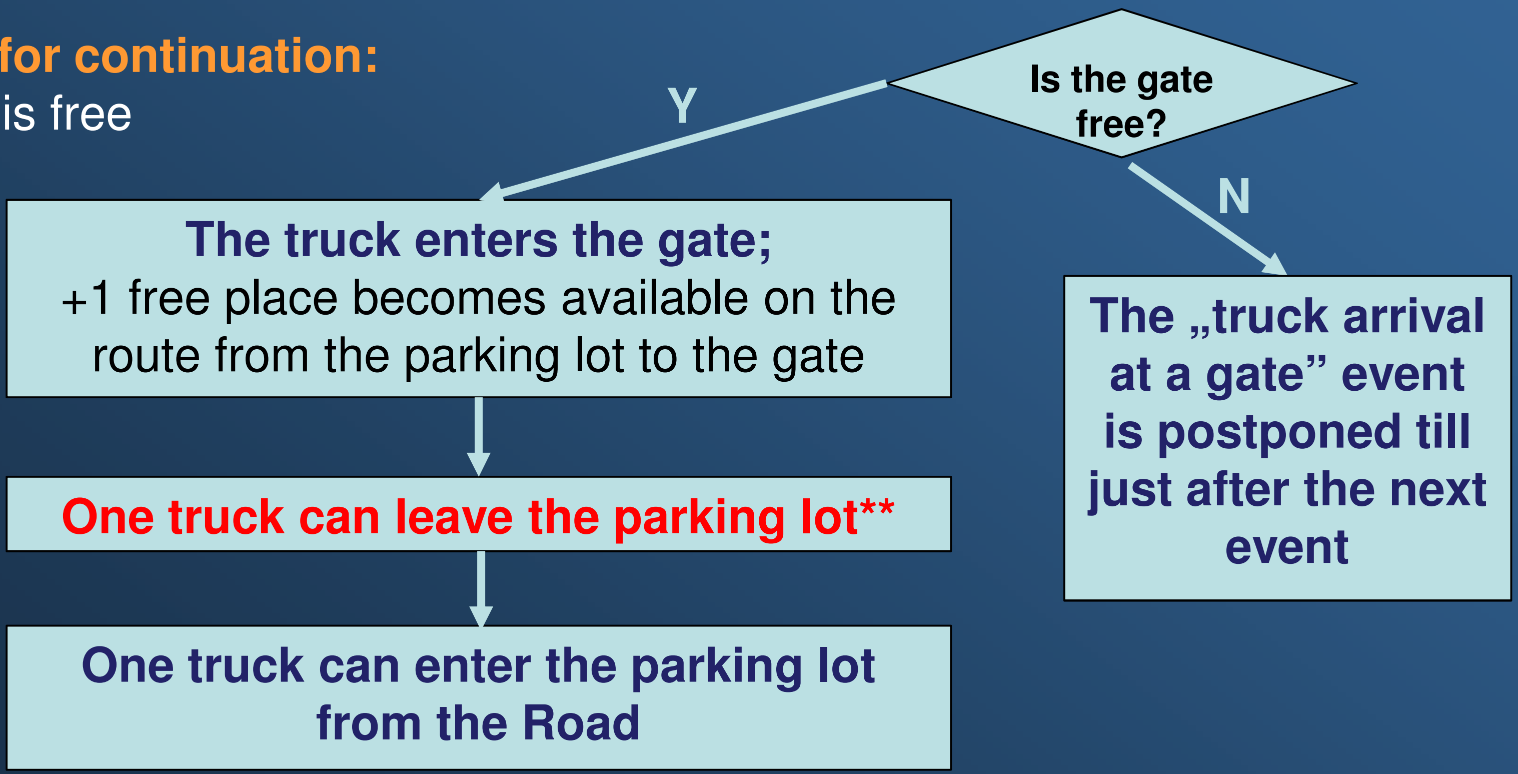
Constraint:

- capacity: 1

Criteria for continuation:
the gate is free

Event: Arrival of a truck at a gate

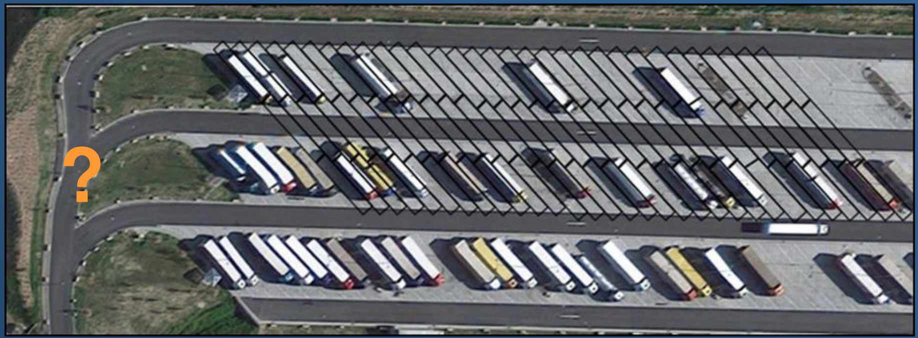
Event handling:



Operation of the simulation framework
The Gate

One truck can leave the parking lot**

Decision point: which truck to choose to leave the parking lot?



Applied algorithms:

1. Select a truck from the parking lot – amongst the vehicles that have every condition fulfilled to start to their destination - **randomly**.

2. Choose the truck **with the closest deadline** from from the parking lot – amongst the vehicles that have every condition fulfilled to start to their destination

3. Select the truck **with minimal overall cost for the truck** from from the parking lot – amongst the vehicles that have every condition fulfilled to start to their destination

Operation of the simulation framework
Route from gate to dock

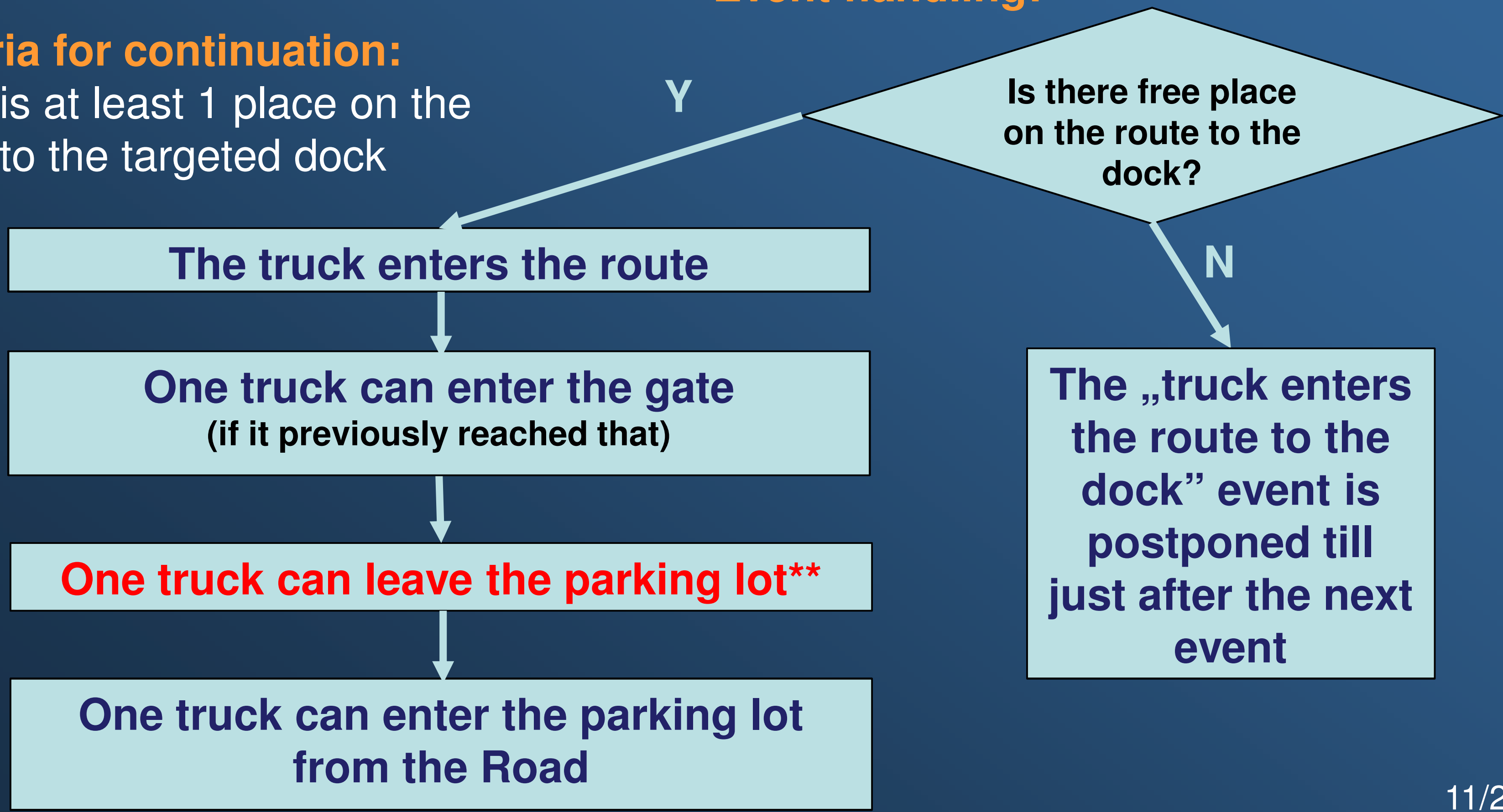
Constraints:

- capacity: $c_{i,j}$
- works in FIFO manner

Criteria for continuation:
there is at least 1 place on the route to the targeted dock

Event: A truck enters the route from the gate to the desired dock

Event handling:



Operation of the simulation framework
The Dock - arrival

Constraint:

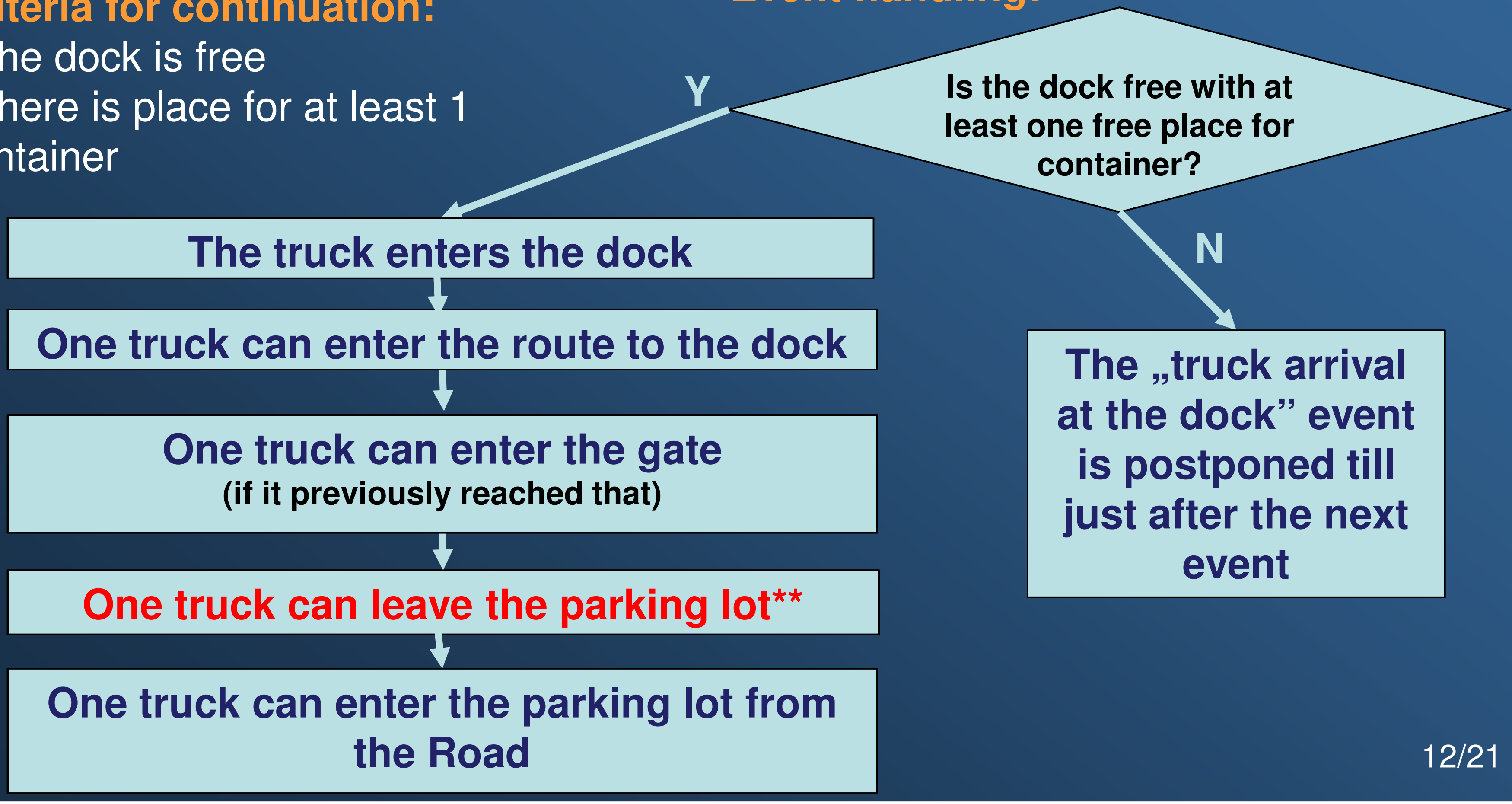
- capacity: 1

Criteria for continuation:

- the dock is free
- there is place for at least 1 container

Event: Arrival of a truck at a dock

Event handling:



Operation of the simulation framework
The Dock - unload

Constraints:

- Capacity for truck: 1
- Capacity for container: $c^{container}_j$

Criteria for continuation: -

Event: Truck unloaded at the dock

Event handling:

The truck „disappears" from the system

↓

One truck can enter the dock

↓

One truck can enter the route to the dock

↓

One truck can enter the gate
(if it previously reached that)

↓

One truck can leave the parking lot**

↓

One truck can enter the parking lot from the Road

Operation of the simulation framework
The Ship-related part

Constraints:

- Capacity for ships: 1
- Capacity for container: $c^{container}_j$

Event: A ship arrives at the port

Event handling:

Is there 0 ship in the dock AND is there at least one container for the ship?

Y → The ship enters the dock

N → The „ship arrival at the dock" event is postponed till just after the next event (the ship waits for its dock)

The structure of the input data 1/2

(related to the trucks / ships)

- number of trucks: **T**
- number of ships: **S**
- ID of the m^th truck: id^T_m, $m \in [1, \ldots, T]$
- ID of the n^th ship: id^S_n, $n \in [1, \ldots, S]$
- Arrival time of the m^th truck at the port: T^A_m, $m \in [1, \ldots, T]$
- Deadline for the m^th truck to leave the port: T^B_m, $m \in [1, \ldots, T]$
- Arrival time of the n^th ship at the port: S^A_n, $n \in [1, \ldots, S]$
- Deadline for the n^th ship to leave the port: S^B_n, $n \in [1, \ldots, S]$
- The ID of the ship for which the m^th truck carries its container: $ship(id^T_m)$, where $ship(id^T_m) \in \{id^S_n\}, m \in [1, \ldots, T], n \in [1, \ldots, S]$

The structure of the input data 2/2

(related to the trucks / ships)

- Cost of the m^th^ truck for one time unit while stading on the Road: C^{queue}_m
- Cost of the m^th^ truck for one time unit while using it: $C^{TruckOp}_m$
- Cost of the n^th^ ship for one time unit while using it: C^{ShipOp}_n
- Penalty of the n^th^ ship for one time unit while using it after its deadline: $C^{penalty}_n$
- Transfer time of the m^th^ truck on a gate with type $type_i$: $T^{transfer}_{m,i}$, where $i \in [1, \ldots, G]$, $m \in [1, \ldots, T]$
- Ordered list of dock indices for the n^th^ ship: $docks_n$, $n \in [1, \ldots, S]$, and each element of $docks_n$ is between 1 and D.

Example for an input data (parts)

(related to the trucks / ships)


```
{
  "numberOfTrucks":100,
  "numberOfShips":20,
  "Trucks":[
    {
      "id":1,
      "arrivalDatetime":"2023-05-08 20:48",
      "shipId":16,
      "queueCost":15,
      "vehicleCost":6,
      "gateTypeTransferTimes":{
        "A":1,
        "B":5
      },
      "deadline":"2023-05-08 22:49"
    },
    {
      "id":2,
      "arrivalDatetime":"2023-05-07 07:52",
      "shipId":19,
      "queueCost":19,
      "vehicleCost":20,
      "gateTypeTransferTimes":{
        "A":8,
```

```
"Ships":[
  {
    "name":1,
    "earliestDatetime":"2023-05-11 12:47",
    "latestDatetime":"2023-05-12 08:34",
    "dockIndex":[
      0
    ],
    "shipCost":27,
    "penaltyAfterDeadline":387
  },
  {
    "name":2,
    "earliestDatetime":"2023-05-10 19:24",
    "latestDatetime":"2023-05-11 17:09",
    "dockIndex":[
      3
    ],
    "shipCost":142,
    "penaltyAfterDeadline":214
  },
```

The examined input classes

For all the inputs, the structure of the port and the number of the vehicles were the same: **G = 3, D = 4, T = 100, S = 20**

The input classes differ in the **time windows** of the trucks and the ships.

1st input class

Trucks:

Arrival: now + random(0-3 days)

Deadline: arrival + random(1-10 hours)

Ships:

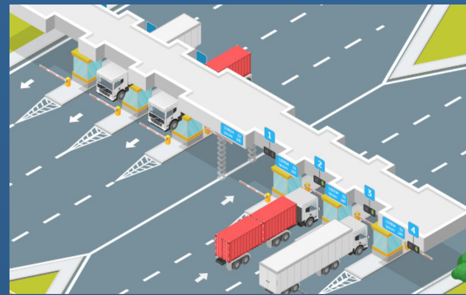
Arrival: now + random(0-3 days)

Deadline: max(last truck's arrival to it; arrival) + 1 hour

2nd input class: Extended deadlines: by 3 days

The applied algorithm variants

The two decision points:



1. To which gate to direct the arrived truck?

1.1. to the gate that results in **minimum overall cost** for the truck

1.2. to a **random** gate, from which the destination dock can be reached

2. Which truck to select from the parking lot to start?

2.1. the one with the **closest deadline**

2.2. the one with the **mimimum overall cost**

2.3. select the truck **randomly**

Results

Target function: **minimum overall cost** of the trucks/ships

3 x 2 x 2 x 10 runs

Results on input class1 (class2 – ext. deadline)

Truck-selection policy from the parking lot

Gate-selection policy

	Closest deadline	Minimum cost	Random
Minimum cost	- 79 (87)% overall cost related to the worst found on average - 16 (1)% deadline overrun	- **72 (73)% overall cost related to the worst found on average** - 42 (2)% deadline overrun	- 89 (89)% overall cost related to the worst found on average - 36 (4)% deadline overrun
Random	- 85 (86)% overall cost related to the worst found on average - **13 (0)% deadline overrun**	- 83 (85)% overall cost related to the worst found on average - 30 (0)% deadline overrun	- 97 (92)% overall cost related to the worst found on average - 40 (2)% deadline overrun

Summary

An event-based simulation framework was developed that can handle cutomized port structures

2 decision points were identified (gate and truck selection).

3 x 2 approaches were developed for these decision points.

Input data of two input classes were generated.

The behaviour of the different approaches were analyzed on the input by the framework.

Future plans

- **Handling of more decisions (e.g., releasing the FIFO manner)**
- **Analysis of other input classes**
- **Multi-objective target function (or switching between the applied algorithms based on the circumstances)**

Thank you for your kind attention!