COST OPTIMIZATION IN PI CROSS-DOCK NETWORKS USING GENETIC ALGORITHMS

Master 2 Réseaux et système autonomes, paris cité

Presented by Zineb Taieb

Supervisors:
M Hassine Moungla
Mme Essghaier Fatma
M Allaoui Hamid

Outline

01 Introduction

- Context
- Problem statement
- Related work
- Research Gap
- Objective
- Why Genetic Algorithms

02 Genetic Algorithm (GA)

- Genetic Algorithm cycle
- Chromosome design

03 Mathematical model

- Objective function
- Fitness function
- Constraints

04 Genetic operators

- Genetic operators
- Unfeasibility
- Handling unfeasibility

05 Experimental study

- Benchmark
- Computational setup
- Comparison
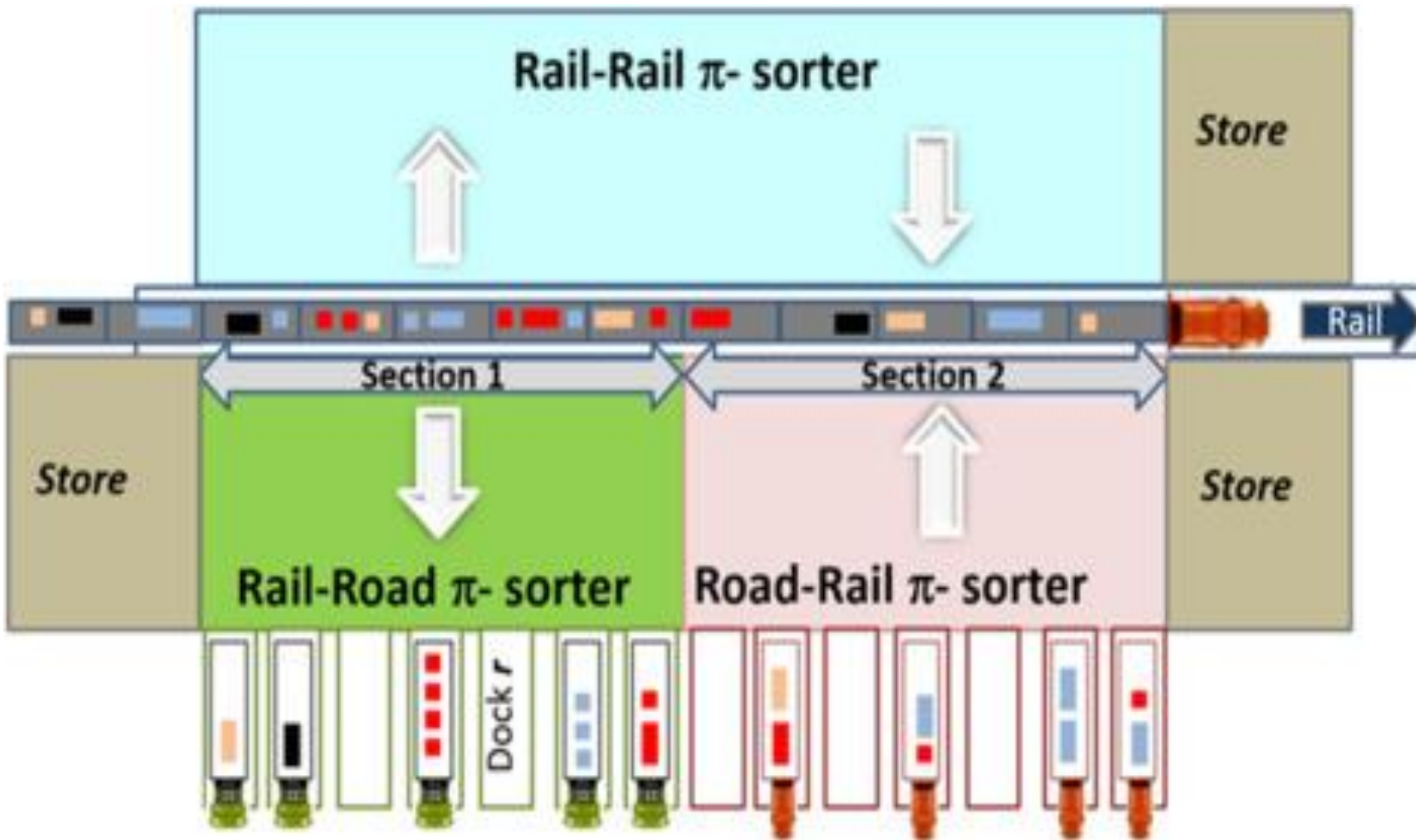
04 Conclusion and future work

Definition

Cross-docking is a logistics approach that enables fast transshipment of goods between inbound and outbound transportation modes with minimal storage. Due to the involvement of multiple docks, modes, and constraints, optimizing cross-dock operations is a challenging problem that requires efficient scheduling and assignment strategies.
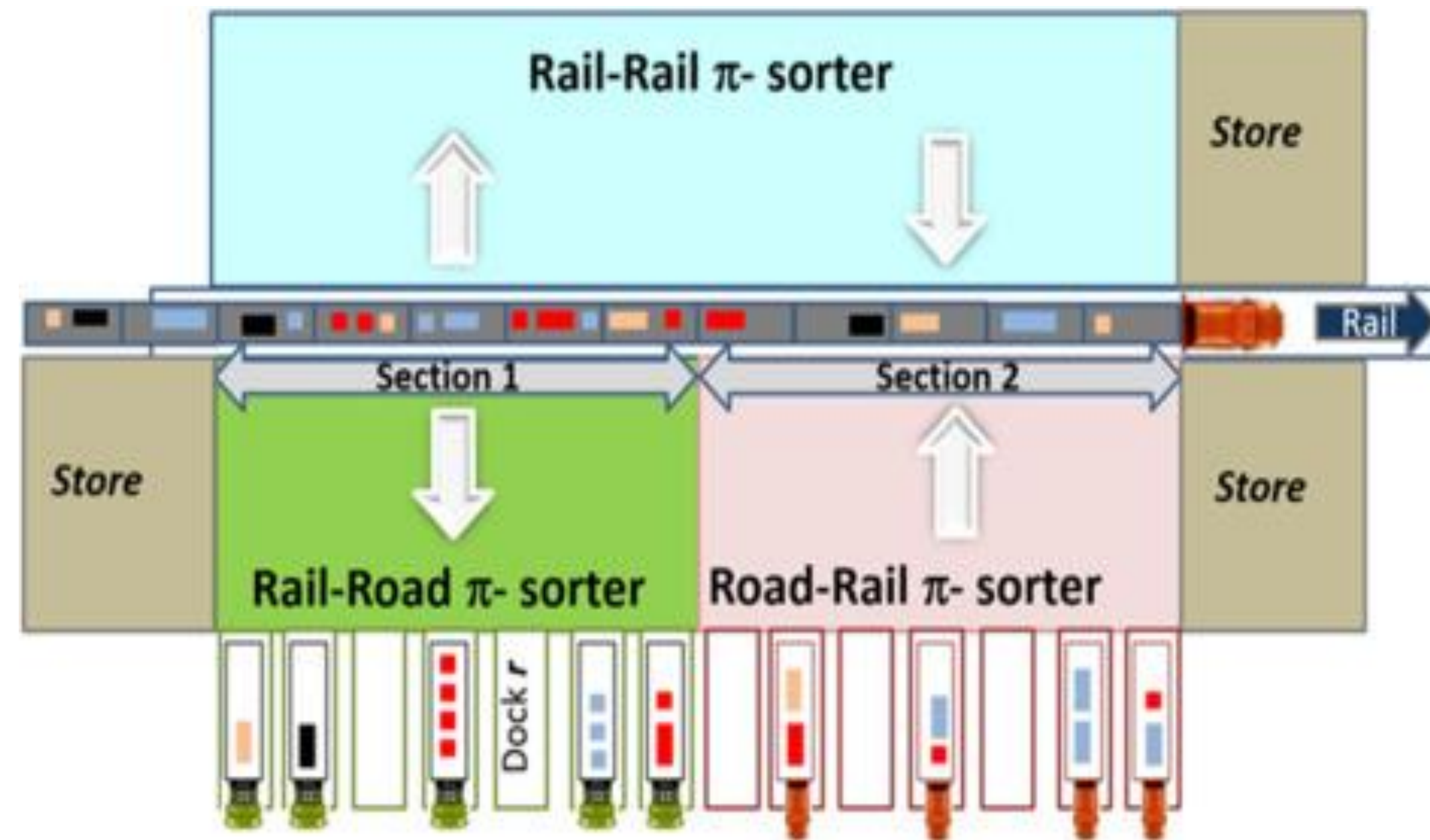
Physical Internet cross dock sorter [1]

Problem statement

How to efficiently schedule and assign containers, vehicles, and docks in a cross-docking platform so as to minimize cost and energy consumption under strict operational constraints and limited computational time.

Physical Internet cross dock sorter [1]

Reference	Objective Function	Methods
Chargui, et al. (2019) Truck Scheduling in a Rail–Road PI cross-dock Considering Energy Consumption	Min energy consumption + vehicle costs	Hybrid metaheuristic (Variable Neighborhood Search+ Tabu Search)
Shahram, et al. (2019)Trucks scheduling in cross-docking with energy consumption consideration and trucks queuing	Min costs of holding products in a cross-dock + energy consumption of forklift in a cross-doc	Multi objective imperialist competitive algorithm (MOICA) + multi objective grey wolf optimizer (MOGWO)
Aberka, et al. (2024) Sustainable Multi-Objective Truck Scheduling in a Rail-Road Physical Internet Cross-Docking Hub with Internal Storage	Min storage cost and energy consumption	Multi-objective Mixed Integer Programming Model (MO MIP) SEPLEX (exact method)
Madani-Isfahani, et al. (2014), Multiple cross-docks scheduling using two meta-heuristic algorithms	Min operation time	Simulated Annealing(SA) + Firefly Algorithm
Yu, et al. (2021), Hybrid GA Truck Scheduling and Product Routing on the Cross-Docking System	Min completion time	Hybrid GA
Mohtashami.A (2015) Scheduling Trucks in Cross Docking Systems with Temporary Storage and Repetitive Pattern for Shipping Trucks	Min completion	Genetic Algorithm
Maxim A. Dulebenets (2018), A Diploid Evolutionary Algorithm for Sustainable Truck Scheduling at a Cross-Docking Facility	Min total cost	Evolutionary Algorithm

Research gap

Some works focus on:

- Energy consumption (Chargui et al. (2019), Shahram et al. (2019))
- Completion time (Madani-Isfahani et al ,(2014), Yu, Mohtashami et al , (2014))
- Total cost only Dulebenets (2018)

Research gap

Few studies joined

❌ Truck cost

❌ Energy consumption

In a rail–road π-hub layout

Rail-Rail π- sorter

Store

Rail

Section 1   Section 2

Store   Store

Rail-Road π- sorter   Road-Rail π- sorter

Dock r
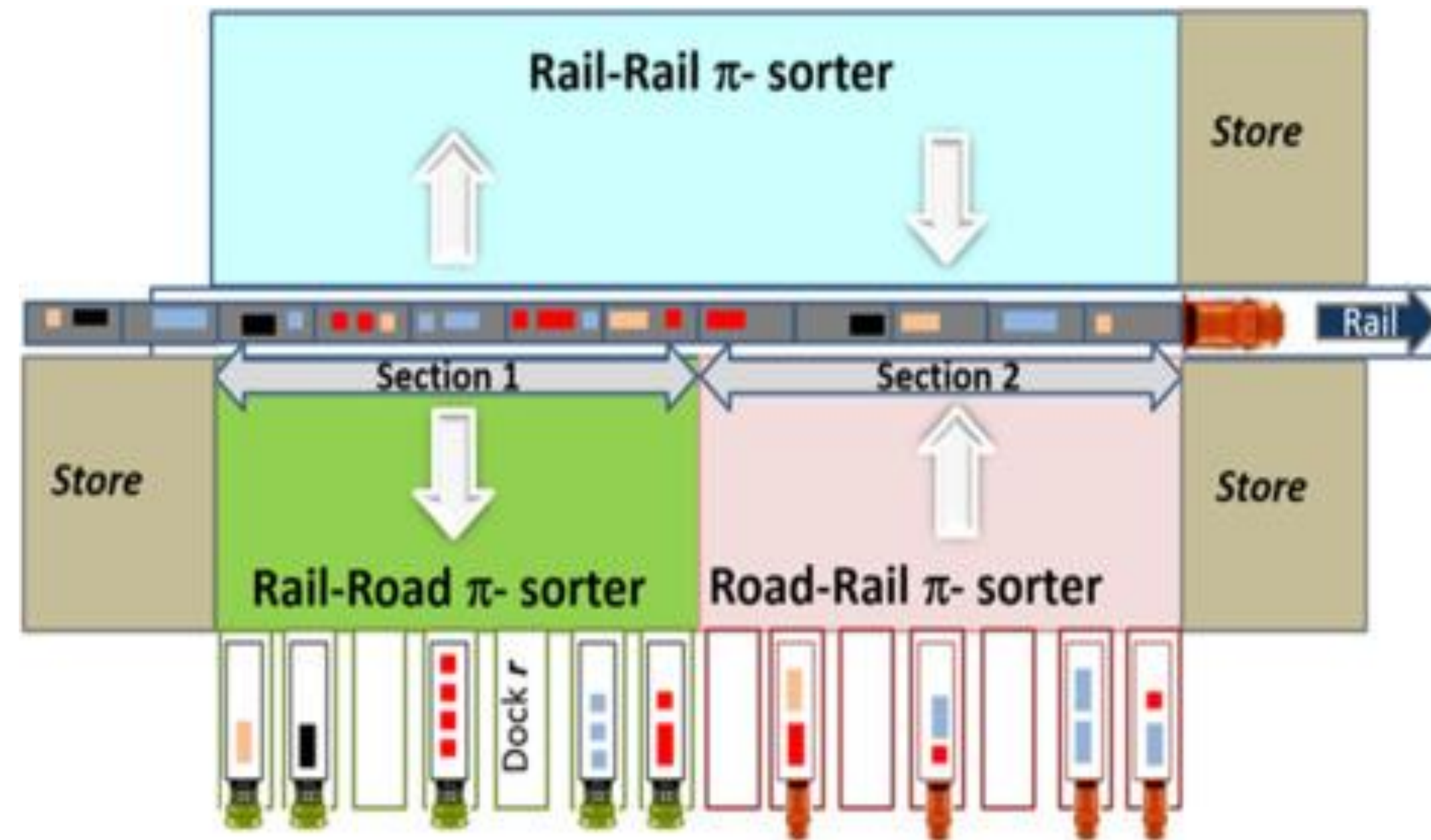
Optimizing：
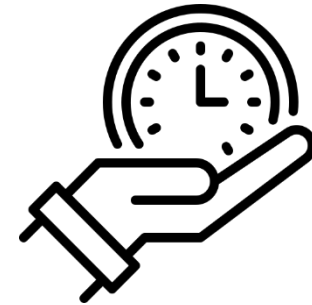
And/or

While :

Completion Time

Cost
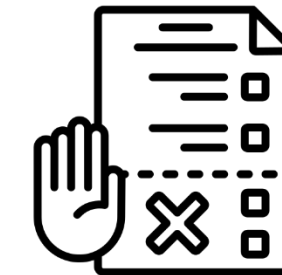
Real world constraints

- NP-hard problem
  Finding the optimal solution in a reasonable time is very difficult.

- Find a near optimal solution with approximative methods
- Explore the search space to find the global optimal solution

Introduction   Genetic operators

Genetic Algorithms (GA)   Experimental study   Conclusion and discussion   • Why GA ?

WHY GA FOR CROSS DOCK OPTIMIZATION?

Objectif

• We aim to optimize container-truck assignments to minimize transport cost and energy cost

NP-hard

• Optimization of this kind of problems is NP-hard

Why Genetic algorithms?

• Robustness, Adaptability, Flexibility(hybridization)

Physical Internet cross dock sorter [1]

Approximative Methods

Problem specific (heuristics)

Meta-heuristics

Nature inspired meta-heuristics

Non-Nature inspired meta-heuristics

- Evolutionary based algorithm

- Swarm Intelligence based algorithm
- Physical/chemical based algorithm

- Tabu search

- Local search

- Large neighbourhood
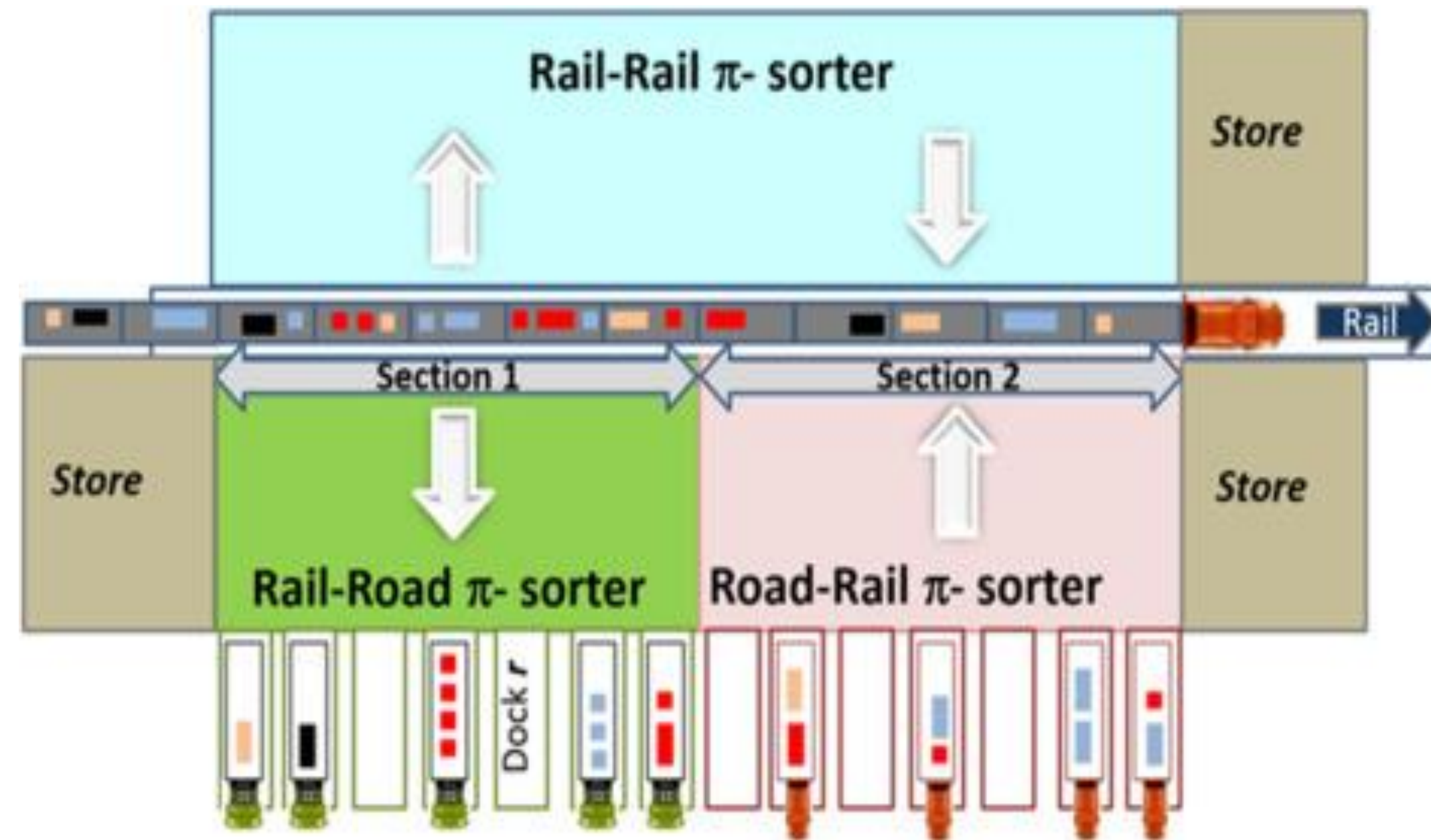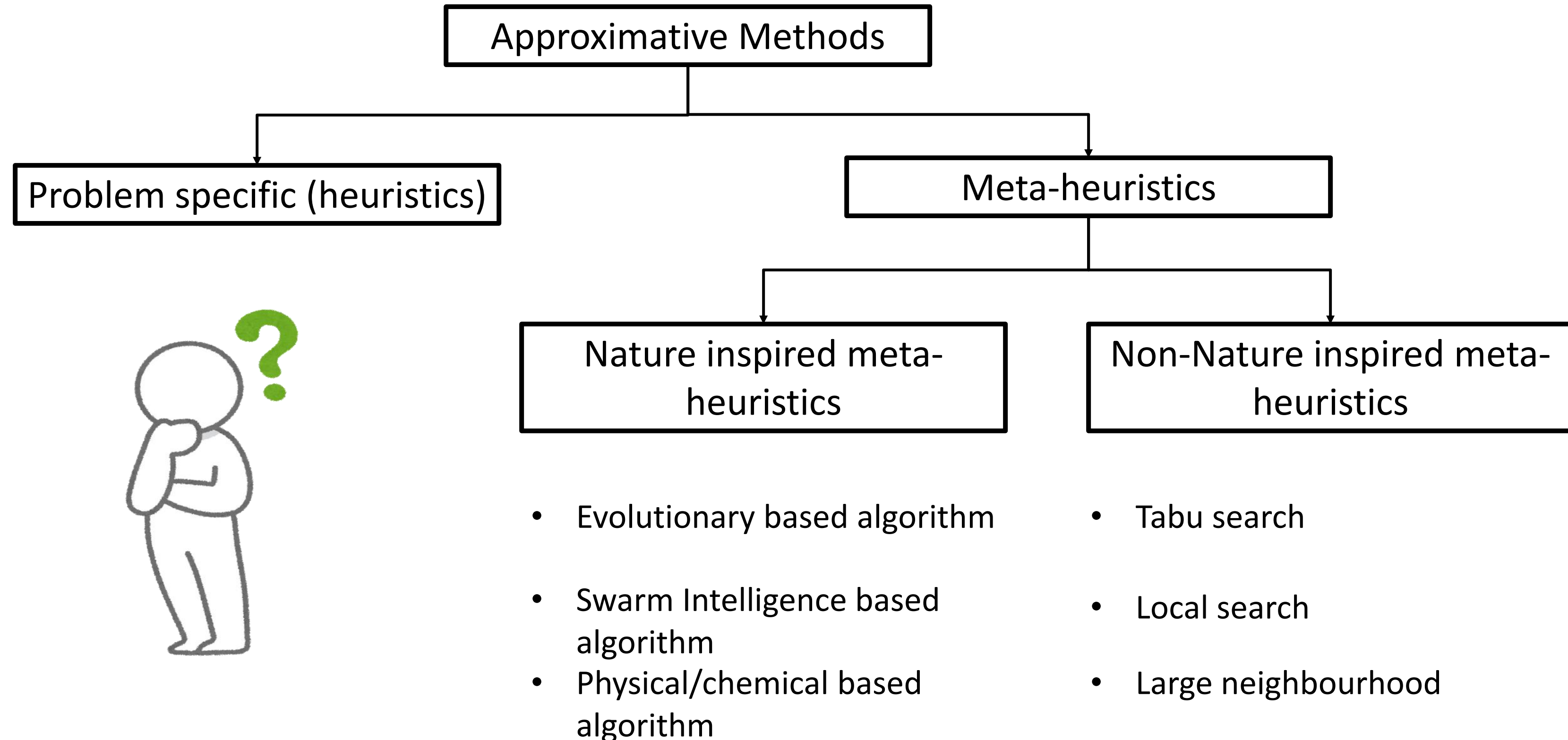
Not Prolem specific

Scalability

Hybridization

Flexibility and adaptability

Integer data representation

Genetic Algorithm Cycle
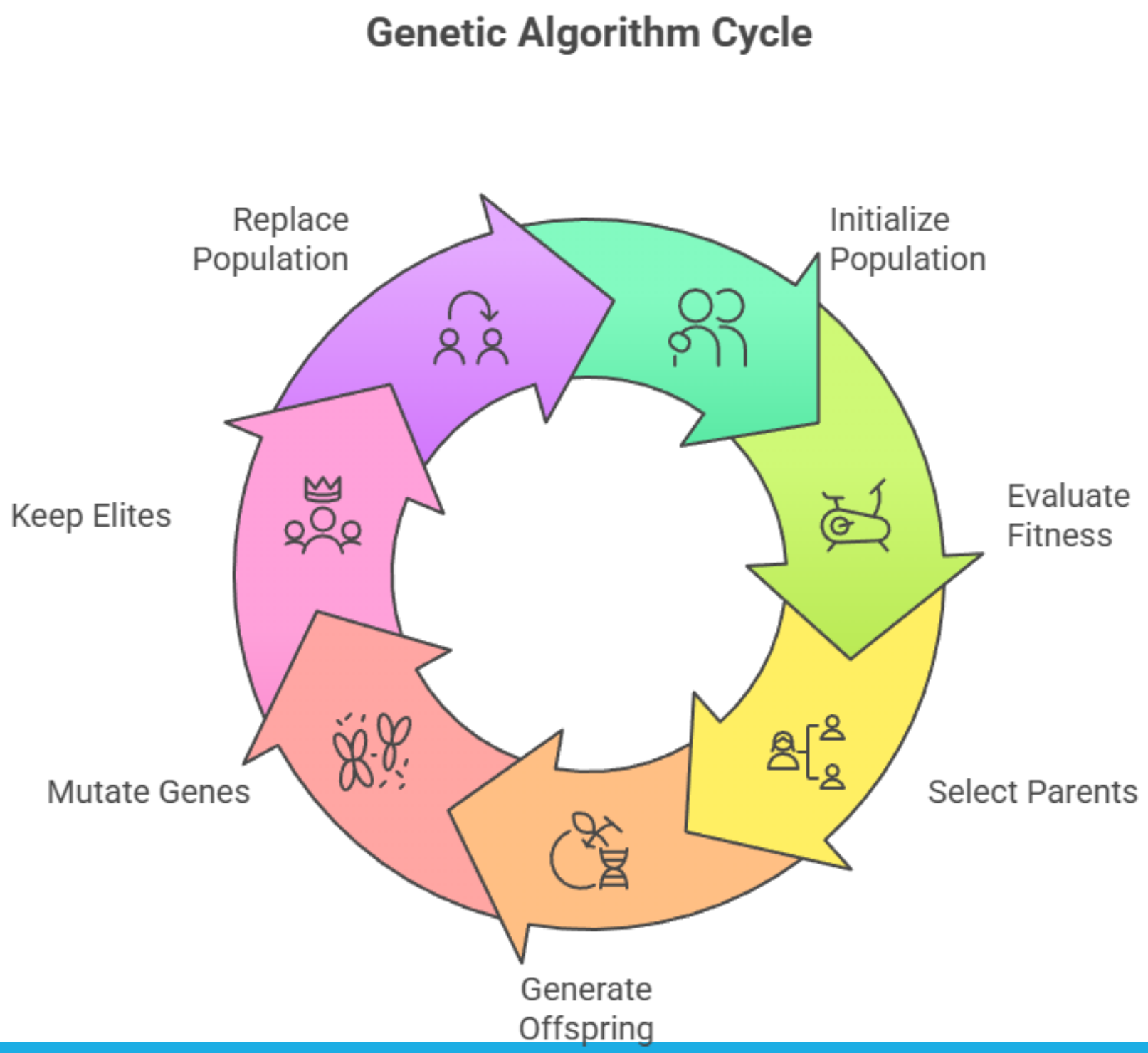
A **chromosome** represents a **solution**, each block represents a truck and its assignments.

| [4] | 0 | 14 | 0 | [ ] | 0 | 2 | 0 | [1,2] | 0 | 4 | 0 | [] | 0 | 10 | 0 |

Truck id : 1
Dock Position: 14
Container : 4

Truck id : 2
Dock Position: 2
Container : none

Truck id : 3
Dock Position : 4
Container : 1 and 2

Truck id : 4
Dock Position : 10
Container : none
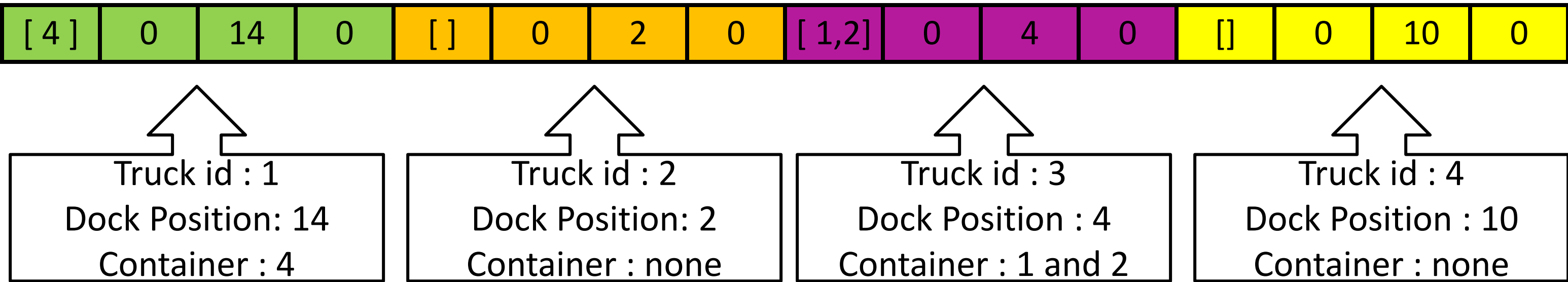
- Truck id : Number to identify a truck
- Dock position : Position of the truck in the Road part of the cross dock hub
- Container : the container id assigned to the truck

Objective 🎯

$$Minimize\ F1 = \sum_{h=1}^{H} \sum_{d=1}^{D} CT_d \times a_{hd}$$

And

$$Minimize\ F2 = CE \times \sum_{i=1}^{N} \sum_{h=1}^{H} z_{ih}$$

[1]

- $F1$: *The cost of using trucks H for destinations D*
- $F2$: *Energy cost to assign container i to truck h*

Parameters :

CT_d : *the cost of using truck h for destination d*

CE : *the cost of one energy unit to transport container i to truck h*

P_i : *the container's i position in the rail area*

R_k : *the truck's h position in the road area*

γ : *vertical length of the cross dock system*

L_i: *Length of the container i*

M : *large number*

Decision variables :

a_{hd} : *binary variable , 1 if truck h is assigned to destination d, 0 otherwise*

P_{ih}: *binary bariable , 1 if container i is assigned to truck h*

x_{hk} : *binary variable , 1 if truck h is assigned to dock k*
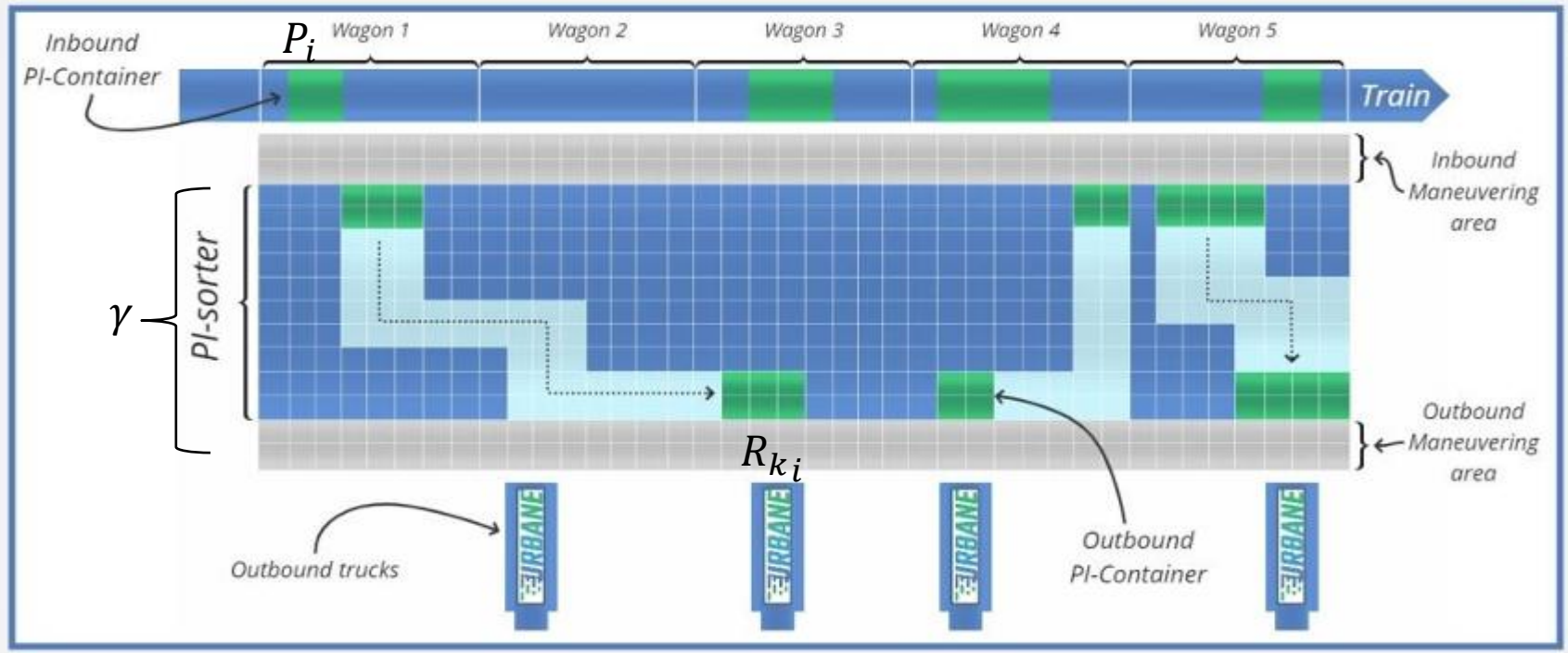
Physical Internet cross dock sorter [1]

$$z_{ih} = 2|P_i - R_k| + \gamma \times L_i - M\big(2 - (p_{ih} + x_{hk})\big)\ \forall\ i, k, h$$

- z_{ih} : *the area swept by the container i to reach the truck h at the k position in the cross dock*

if container i is assigned to truck h and truck h is assigned to dock k in the road s

Genetic Algorithm Cycle
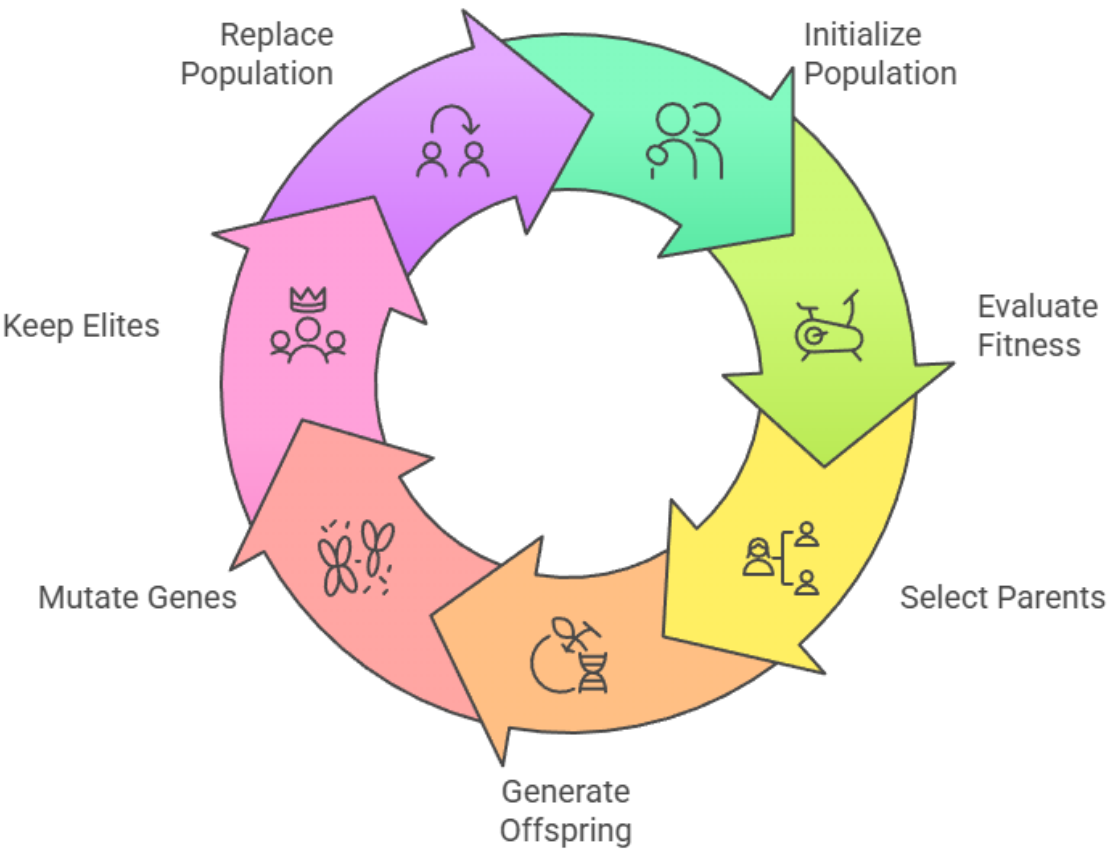
Key step in the Genetic Algorithm loop

Inspired by natural selection process in Evolutionary therory

Evaluation tool

Evaluate Fitness

Evaluation
Shows how to evaluate each solution.

Numerical Value
Assigns a numerical value to a solution.

Objective Function
Fitness function = objective function.

Comparison
Allows comparison and selection of the best solution.

How can we evaluate the fitness of individuals based on two different objectives?

One optimal solution for an objective does not necessarily represent an optimal solution for the second objective.

How can a multi-objective optimization problem (with more than two objectives) be effectively transformed into a single-objective formulation that remains representative of all objectives and suitable for evaluation by a genetic algorithm?

Approach: Weighted–Sum Fitness Function

$$Fitness = W_1 \times F_1 + W_2 \times F_2 + \ldots + W_n \times F_n$$

$$With \sum W_i = 1$$

$$Fitness = W_1 \times F_{1_{Truck\ cost}} + W_2 \times F_{2_{Energy\ cost}}$$

$$W_1 = 0{,}5\ ,\ W_2 = 0{,}05$$

Balancing Multiple Objectives in Optimization

ADVANTAGE | Simplicity

ADVANTAGE | Equal weights = equal influance of F1 and F2 on the total cost

Objective Function F1

Objective Function F2

Weighted sum W1*F1 +W2*F2

Constraints in cross docking operations

Container Assignment — Each container has only one destination

$$\sum_{h=1}^{H} p_{ih} = 1 \; , \forall \, i = 1 \dots N \quad , N \text{ the number of containers}$$

Truck capacity — Containers assigned to truck can not exceed trucks capacity

$$\sum_{i=1}^{N} p_{ih} \times L_i \leq Q, \forall h = 1 \dots H \, , \text{H the number of trucks, Q capacity of truck h ,}$$

L_i the length of container i

Destination consistency

All containers in one trucks must have the same destination

$$p_{ih} + p_{jh} \leq \sum_{d=1}^{D} G_{di} \times G_{dj} + 1 \, , G_{di} \text{ and } G_{dj} \text{ destination matrix,}$$
$$\text{for container and destination}$$

Truck has only one destination and can only load containers with the same destination

$$v_h = \sum_{d=1}^{D} a_{hd} \, , a_{hd} : \text{ binary variable } , 1 \text{ if truck h is assigned to destination d}$$
$$v_h : \text{ binary variable } , 1 \text{ if truck h is used}$$

$$a_{hd} \leq G_{di} + 1 + p_{ih},$$

Truck usage — A truck is used if it has at least one container is assigned to it

$$p_{ih} \leq v_h, v_h \text{ is binary variable}, 1 \text{ if truck h is used } 0 \text{ otherwise}$$

Dock Assignment — A truck can not be at two different docks at the same time

$$\sum_{k=1}^{K} x_{hd} = v_h \, , \; \forall \, k = 1 \; \dots K \, , x_{hd} : \text{binary variable } , 1 \text{ if truck h is assigned to dock k ,}$$

Motivation and context
Genetic Algorithm (GA)

Mathematical formula
Genetic operators

Conclusion and discussion
Experimental study

- **GA operators**
- Unfeasibility

- **Handling Unfeasibility**

Population initialization → Parent selection → Crossover → Mutation → New generation

Hybrid Population Initialization Strategy

- 80% Heuristic-Generated
- 20% Randomly-Generated
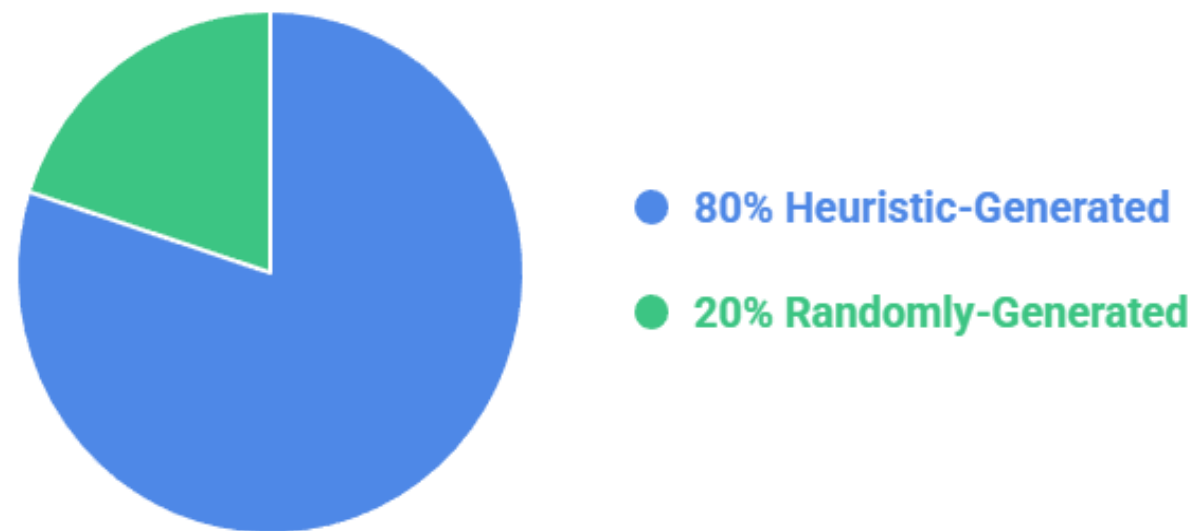
Heuristic population initialization : Binpacking

- Group containers by destination and sort by decreasing length
- For each destination group:
  - If truck is new , assign destination to truck
  - Else assign container to truck with same destination and enough capacity
  - Move to next container

ADVANTAGE Best fit decreasing, fills the trucks to maximum capacity, no truck waste

ADVANTAGE Fast and less computational time

Motivation and context
Genetic Algorithm (GA)
Mathematical formula
Genetic operators
Conclusion and discussion
Experimental study
- GA operators
- Unfeasibility
- Handling Unfeasibility

Population initialization → Parent selection → Crossover → Mutation → New generation
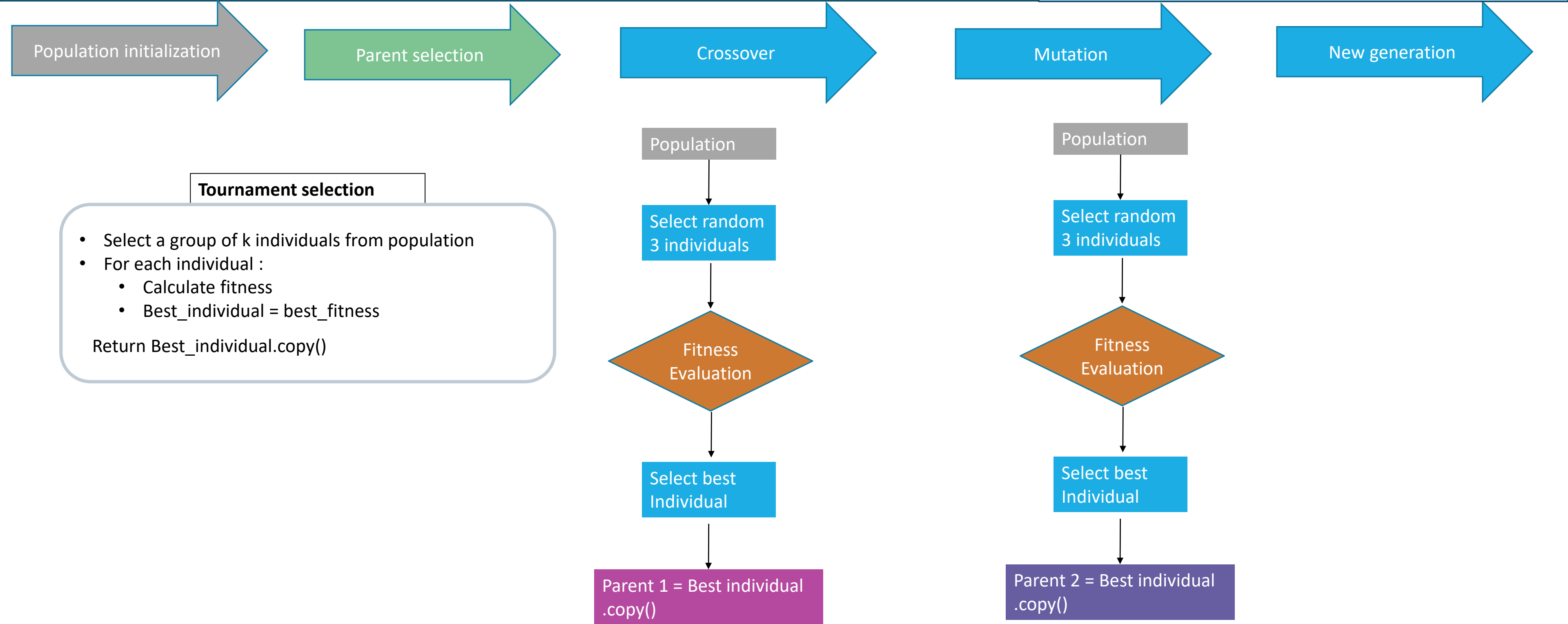
Tournament selection

- Select a group of k individuals from population
- For each individual :
    - Calculate fitness
    - Best_individual = best_fitness

Return Best_individual.copy()

Population
↓
Select random 3 individuals
↓
Fitness Evaluation
↓
Select best Individual
↓
Parent 1 = Best individual .copy()

Population
↓
Select random 3 individuals
↓
Fitness Evaluation
↓
Select best Individual
↓
Parent 2 = Best individual .copy()

Motivation and context
Genetic Algorithm (GA)

Mathematical formula
Genetic operators

Conclusion and discussion
Experimental study

- **GA operators**
- Handling Unfeasibility
- Unfeasibility

Population initialization → Parent selection → Crossover → Mutation → New generation

Parent 2 : [[3,7],0,4 ,0,{Crossover point} [],0,7 ,0,[2],0, 6, 0]

Parent 1 : [[4,6],0,3,0, {Crossover point}[1],0, 5 ,0,[],0, 1 ,0]

Offspring 1 : [[4,6],0,3,0, [],0,7 ,0,[2],0, 6, 0]

Offspring 2 : [[3,7],0,4 ,0, [1],0, 5 ,0,[],0, 1 ,0]
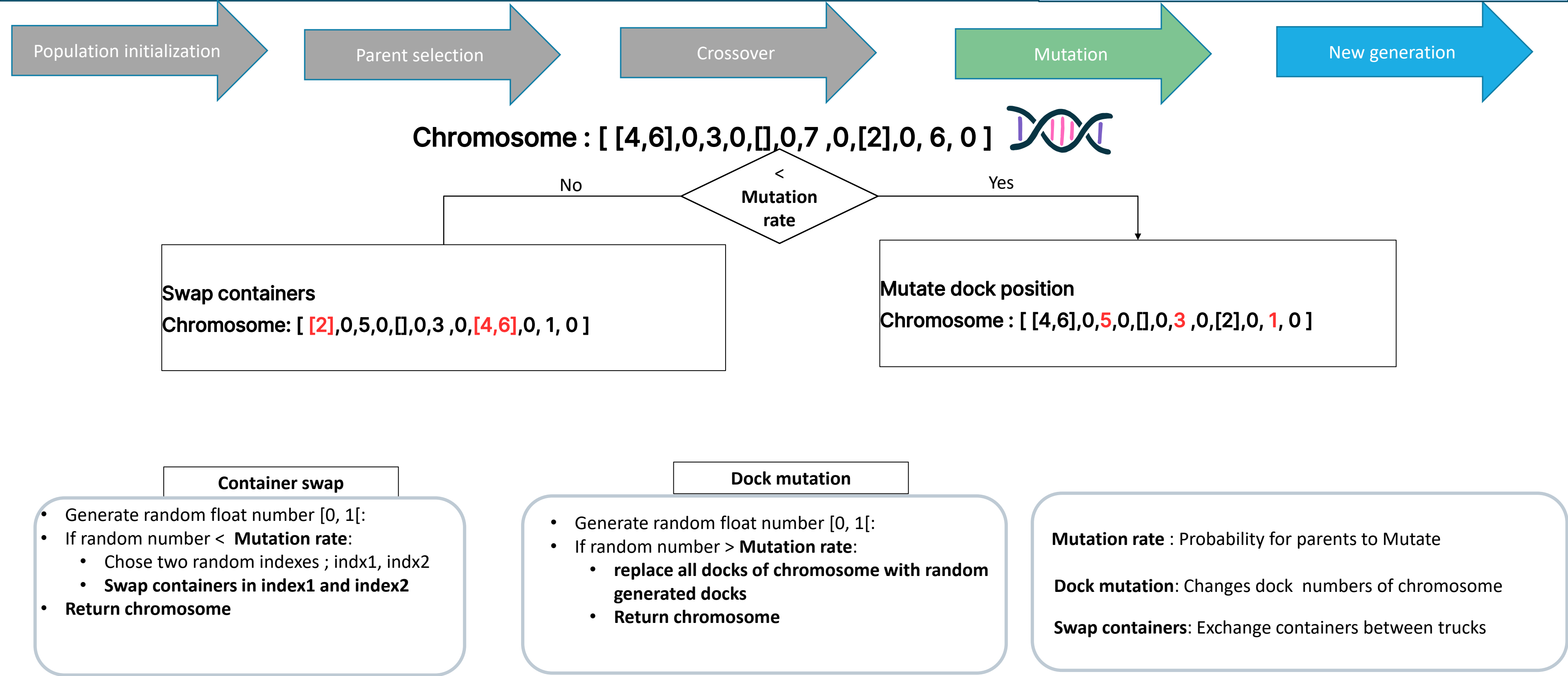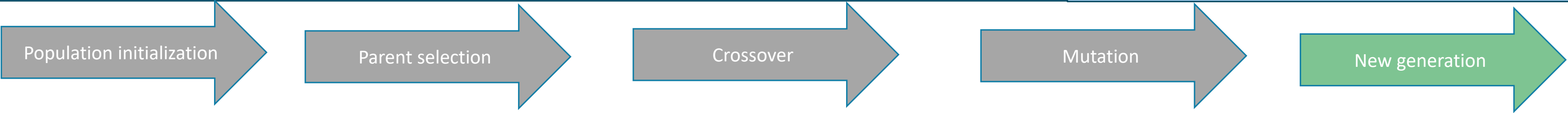
One point Crossover

- Generate random float number [0, 1[:
- If random number< **mutation rate**:
    - **One point cross over (Parent 1, Parent 2)**
- Else :
    - **Offspring 1, Offspring 2 = copy(Parent 1, Parent 2)**

Crossover rate : Probability for parents to crossover

One point crossover : Preserves the chromosome structure

Motivation and context
Genetic Algorithm (GA)

Mathematical formula
Genetic operators

Conclusion and discussion
Experimental study

- **GA operators**
- **Handling Unfeasibility**
- Unfeasibility

Population initialization → Parent selection → Crossover → Mutation → New generation

Chromosome : [[4,6],0,3,0,[],0,7 ,0,[2],0, 6, 0]

< Mutation rate

No

Yes

Swap containers
Chromosome: [[2],0,5,0,[],0,3 ,0,[4,6],0, 1, 0]

Mutate dock position
Chromosome : [[4,6],0,5,0,[],0,3 ,0,[2],0, 1, 0]

Container swap

- Generate random float number [0, 1[:
- If random number < **Mutation rate**:
    - Chose two random indexes ; indx1, indx2
    - **Swap containers in index1 and index2**
- **Return chromosome**

Dock mutation

- Generate random float number [0, 1[:
- If random number > **Mutation rate**:
    - **replace all docks of chromosome with random generated docks**
    - **Return chromosome**

Mutation rate : Probability for parents to Mutate

Dock mutation: Changes dock numbers of chromosome

Swap containers: Exchange containers between trucks

| Motivation and context | Mathematical formula | Conclusion and discussion | • GA operators | • Handling Unfeasibility |
| Genetic Algorithm (GA) | Genetic operators | Experimental study | • Unfeasibility | |

Population initialization → Parent selection → Crossover → Mutation → New generation

Mutation & Crossover can generate unfeasible solutions

Example

Chromosome	Violation of the constraints ⚠
[[3,2,1],0,3,0, [4,2],0, 5 ,0,[],0, 1 ,0 ,[1,2],0,10,0,[],0,1,0]	• **Multiple destination** in truck 1 • Container **duplicated**: 1,2 • Truck 1 :[3,2,1] → destination : 1, container 2 →destination :2 →**destination mismatch**
[[],0,14,0, [],0, 2 ,0,[],0, 1 ,0 ,[],0,11,[],0,10,0]	• **Empty trucks**
[[4,2,3,1,5],0,14,0, [],0, 4 ,0,[],0, 1 ,0 ,[],0,13,[],0,10,0]	• **Truck 1 capacity exceeded**, length(4,2,3,1,5) = 15> truck's capacity(13)

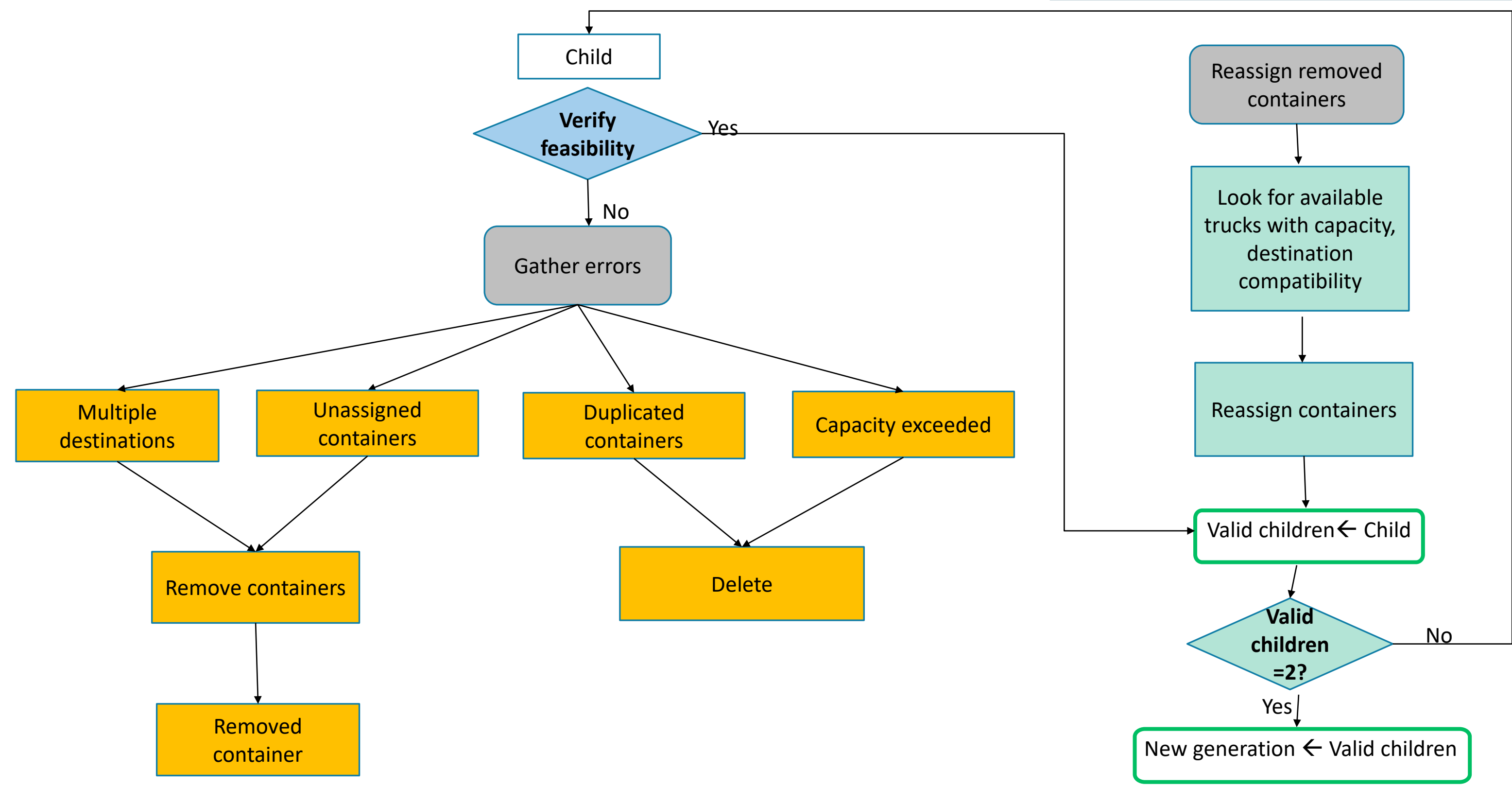Score very low fitness = considered "good solution" for the GA ➡ "good solutions" survive to the next generation

Good new ! Prevent unfeasible solutions from propagating by a simple strategy

Truck id	Destination	Capacity
1	1	13
2	2	13
3	1	13
4	2	13

Container id	Destination	Length
1	2	3
2	2	5
3	1	2
4	3	3
5	4	4

Motivation and context   **Mathematical formula**   **Conclusion and discussion**   • GA operators   • **Handling Unfeasibility**

Genetic Algorithm (GA)   **Genetic operators**   **Experimental study**   • **Unfeasibility**
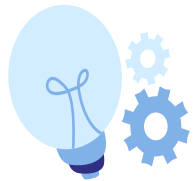
Strategy: repair wrong chromosomes, implemented in the correct_chrom function

[[3,2,1],0,3,0, [4,2],0, 5 ,0,[],0, 1 ,0 ,[1,2],0,10,0,[],0,1,0]

Gather errors

[3,2,1]

Multiple destinations: (2,1)

Keep containers with dominant destination, remove others

[3X,1]

Removed container : 2

[1,2]   [4,2]

Duplicated containers : 1,2

Remove containers (1,2)

[X,X,   [4,X]

Reassign removed containers

Removed = 2

Look for available trucks with capacity, destination compatibility

Truck 1 → destination 1 ✖
Truck 2 → destination 2 ✔

[[3,1],0,3,0, [4,2],0, 5 ,0,[],0, 1 ,0 ,[],0,10,0,[],0,1,0]

Objective:

Compare the performance of different approaches on the same dataset using benchmark_set_for_sustainability_2019 by Chargui

Implementation method :

- GA (Controlled + Random Population): mix of random and guided initialization.
- Exact Method: used as a benchmark for optimal results.

Evaluation method :

- All methods executed on the same set of instances
- Measured cost per instance to evaluate performance.
- Results presented in plots comparing cost for three methods

Instances taken from benchmark available in the link:

Symbol	Description	Meaning
N	Number of containers	Total number of containers to be handled
D	Number of destinations	Total number of distinct destinations
H	Number of trucks	Total number of available trucks
K	Number of docks	Number of loading/unloading docks
Q	Truck capacity	Maximum capacity of each truck
C_e	Energy cost per unit	Energy cost per unit of distance
Cdt	Truck cost for destination d	Fixed cost of using a truck assigned to destination d
Y	Vertical length of the cross-dock	Vertical dimension of the cross-dock layout
L	Container length	Length (or size) of containers
P	Container positions	Positions of containers inside the cross-dock
R	Truck positions	Positions of trucks at the docks
G	Destination–container matrix	Binary matrix indicating the destination of each container

```
/////////////////////////////////////////////////
// Inst_4_1_4
/////////////////////////////////////////////////

N       =           4;
K       =           15;
D       =           1;
H       =           4;
Q       =           13;

C_e = 0.5;

C_d = [
            351
];

I = 10;

Y = 12;

V = 5;

L = [
            3           5           2           3
];

G = [
            [1          1           1           1]
];

P = [
            11          27          42          54
];

R = [
            3           8           13          18          23
            28          33          38          43          48
            53          58          63          68          73
];
```

Instances taken from benchmark available in the link: https://www.researchgate.net/publication/333592362_Multi-Objective_Sustainable_Truck_Scheduling_in_a_Rail-Road_Physical_Internet_Cross-Docking_Hub_Considering_Energy_Consumption

Symbol	Description	Meaning
N	Number of containers	Total number of containers to be handled
D	Number of destinations	Total number of distinct destinations
H	Number of trucks	Total number of available trucks
K	Number of docks	Number of loading/unloading docks
Q	Truck capacity	Maximum capacity of each truck
C_e	Energy cost per unit	Energy cost per unit of distance
Cdt	Truck cost for destination d	Fixed cost of using a truck assigned to destination d
Y	Vertical length of the cross-dock	Vertical dimension of the cross-dock layout
L	Container length	Length (or size) of containers
P	Container positions	Positions of containers inside the cross-dock
R	Truck positions	Positions of trucks at the docks
G	Destination–container matrix	Binary matrix indicating the destination of each container

Table 1 : Description of instance data

Instance	file Name	containers	Destinations	Docks
4	inst_4_3_4.txt	4	1-2-3	15
5	inst_5_2_4.txt	5	1-2-3	15
6	inst_6_3_5.txt	6	1-2-3	15
7	inst_7_1_5.txt	7	1-2-3	15
8	inst_8_1_5.txt	8	1-2-3	15
9	inst_9_2_7.txt	9	1-2-3	15
10	inst_10_1_7.txt	10	1-2-3	15
11	inst_11_2_7.txt	11	1-2-3	15
12	inst_12_3_7.txt	12	1-2-3	15
20	inst_20_7_15.txt	20	7	15
20	inst_20_10_15.txt	20	10	15
30	inst_30_7_20.txt	30	7	15
30	inst_30_10_20.txt	30	10	15
30	inst_30_15_20.txt	30	15	15

Model is **NP-hard** → requires **higher computational time and resources**
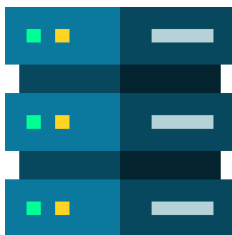
LGI2A provided access to a HPC(High-performance Computing) cluster for running complicated calculations without disruption

Why Computational cluster ?

Mixed Integer models and iterative algorithms require significant CPU time

32 large instances, complicated calculations lead high computational complexity

HPC : scalable and high computational power, stable long execution without interruptions

Table 2 : Description of Cluster Tech3E

Company	Name	Access	Usage	CPU	Memory	Link
HPE Hewlett-Packard Enterprise	Tech3E (LGI2A _ Université d'Artois)	Slurm Workload manager, Linux	Batch execution of large experiments	2 x AMD EPYC 9224 CPU , 24 cores , 2 threads per core	640 Go theory, 629 Go available	https://www.hpe.com/emea_europe/en/hpe-proliant-dl385-gen11.html

Gurobipy: A gurobi python interface, version : python3.12-gurobi

Gurobi is a Mathematical solver , for decision making problems

Optimize :

$$Minimize\ F1 = \sum_{h=1}^{H}\sum_{d=1}^{D} CT_d \times a_{hd}$$

And

$$Minimize\ F2 = CE \times \sum_{i=1}^{N}\sum_{h=1}^{H} z_{ih}$$

- $F1 : The\ cost\ of\ using\ trucks\ H\ for\ destinations\ D$

- $F2 : Energy\ cost\ to\ assign\ container\ i\ to\ truck\ h$

Unified Objective function: Weighted Sum, W1=0.5 W2= 0.5

$$Fitness = W_1 \times F1_{Truck\ cost} + W_2 \times F2_{Energy\ cost}$$

$W_1 = 0,5\ ,W_2 = 0,05$

Exact method code structure

```
//Set Parameters
N_list # list of containers
H_list # list of trucks
K_list # list of docks
D_list # destinations

W1, W2 = 0.5, 0.5
Y = 12 # length of cross dock system

// initialize model

m = Model(f"Instance_{instance_id}_Transport_Energie« )
# Time limit: 2 hours
m.setParam("TimeLimit", 7200)

//Add decision variables
a : binary d is destination of the truck h
X : binary truck h is assigned to dock k
P : binary container n is assigned to truck h
v_used : binary truck is used ,
n : binary same dock for two trucks
Z : continous area swept by the container

//Add constraints
m.addConstrs
.
.
//Solve
m.optimize()

//Save results in csv
df_results = pd.DataFrame(results)
df_results.to_csv("results_exact_summary_with_chargui_instances_time
_limit_2_h.csv", index=False)
```

Motivation and context
Genetic Algorithm (GA)

Mathematical formula
Genetic operators

Experimental study

Conclusion and discussion

- Benchmark
- Computational setup

- Comparison

Benchmark_instances_set_for_Sustainability_2019,zip

Table 3 : Exact method parameters

Technical setup	Data	Time limit	Gurobi license
ClusterTech3E	Chargui's benchmark "Benchmark_instances_set_for_sustainability (2019)	Time limits 7200s (2h)	Free academic license, version 12

Table 4 : Approximative method parameters (GA)

Technical setup	Data	Population size	Mutation rate	Crossover rate	Number generations	Heuristic ratio binpacking
Desktop PC	Chargui's benchmark "Benchmark_instances_set_for_sustainability (2019)	50	0,03	0,9	100	0,8

GAP

Instances	Small instances (4 – 5) trucks /containers	Medium instances (6 – 8) trucks /containers	Bigger instances (8 – 12) trucks /containers
GAP	0%	<1%	2-2.5%
interpretation	GA reaches optimal value	GA near optimal value	GAP remains low, GA close to optimal value

Table 5 : Gap (Exact method vs Genetic algorithm)

File Name	Approximative_cost(Truck_cost + Energy_cost)	Exact_cost(truck_cost+energy_cost)	Gap(%)	Computational Time(s)_GA	Computational Time(s)_Exact
inst_4_1_4.txt	243,0	243,0	0	27	0
inst_4_2_4.txt	616,0	616,0	0	27	0
inst_4_3_4.txt	825,0	825,0	0	25	1
inst_5_1_4.txt	474,0	474,0	0	31	0
inst_5_2_4.txt	1009,0	1009,0	0	33	0
inst_5_3_4.txt	1 074,0	1064,0	0.9868421053	42	0
inst_6_1_5.txt	673,0	659,0	2.124430956	37	4
inst_6_2_5.txt	612,0	608,0	0.7401315789	38	0
inst_6_3_5.txt	868,0	800,0	8.5	40	0
inst_7_1_5.txt	741,0	740,0	0.1350438893	42	0
inst_7_2_5.txt	1 022,0	1022,0	0.0489236791	43	0
inst_7_3_5.txt	1 046,0	1045,0	0.1435406699	45	0
inst_8_1_5.txt	1 043,0	1 027,0	1.557177616	55	1
inst_8_2_5.txt	1393,0	1 376,0	1.198692336	51	7
inst_8_3_5.txt	1 424,0	1 400,0	1.713673688	49	6
inst_9_1_7.txt	600,0	600,0	0	59	2
inst_9_2_7.txt	997,0	987,0	1.013171226	59	48
inst_9_3_7.txt	1 340,0	1 336,0	0.2992891882	60	46
inst_10_1_7.txt	966,0	943,0	2.437731849	81	7
inst_10_2_7.txt	1 237,0	1230,0	0.6097560976	83	6
inst_10_3_7.txt	1748,0	1 741,0	0.3732414585	75	232
inst_11_1_7.txt	797,0	793,0	0.5674653216	77	6
inst_11_2_7.txt	1129,0	1 121,0	0.6687472136	73	93
inst_12_1_7.txt	1 223,0	1206,0	1.451077944	1 240	17
inst_12_2_7.txt	1 566,0	1 529,0	2.419091206	68	6

GAP   Total cost   Benchmark comparison   Big instances   Computational Time   Model's Limits

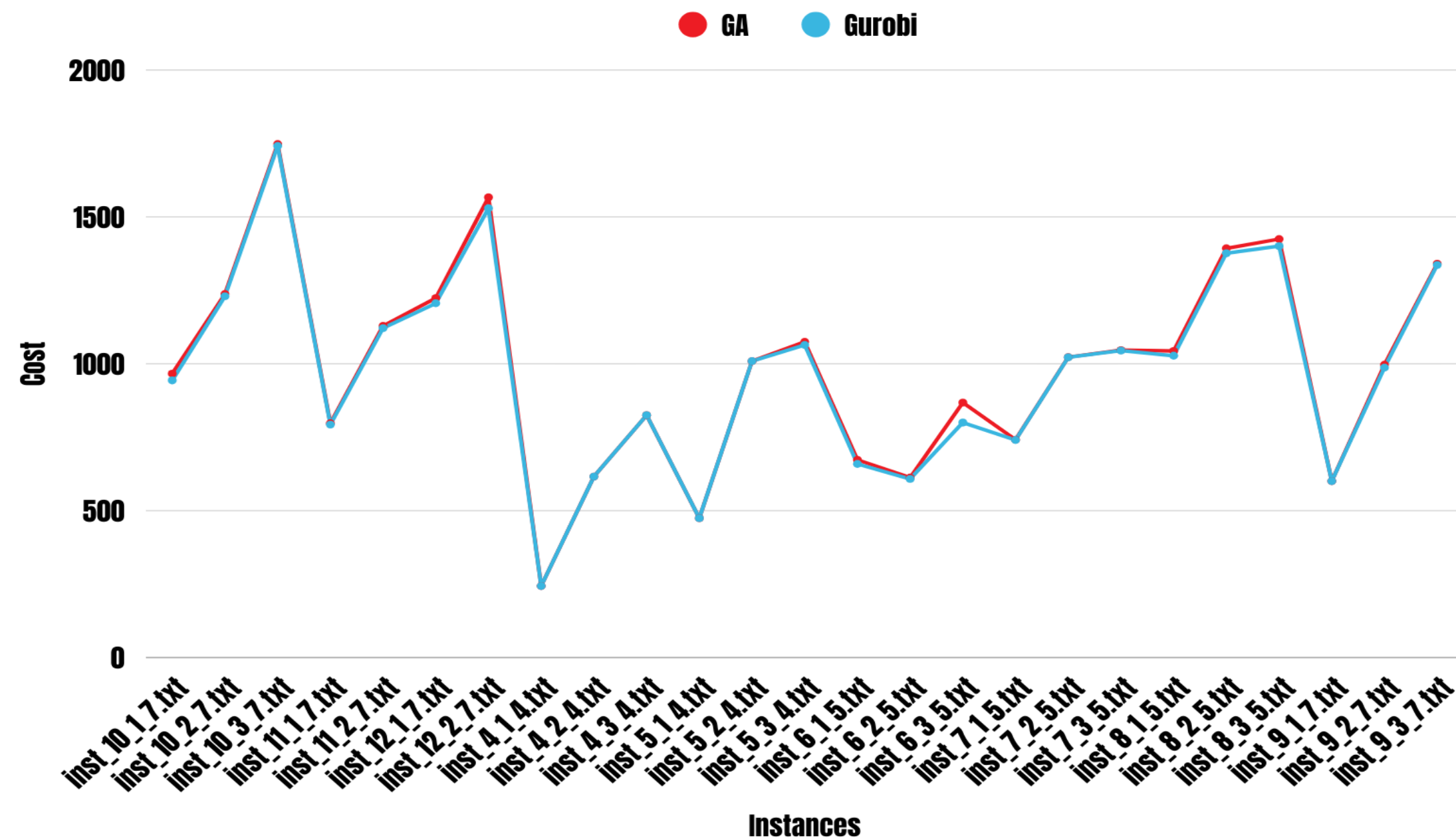GAP   Total cost   **Comparison with benchmark**   Big instances   Computational Time   Model's Limits

- **Truck cost (benchmark H1)** : numerical results given by Chargui et , al (2019), using heuristic (H1) : « Multi-Objective Variable Neighborhood Search hybridized with Simulated Annealing (MO-VNSSA) "

- **Energy cost (benchmark H2)** : numerical results given by Chargui et , al (2019), using heuristic (H2) : « Multi-Objective Variable Neighborhood Search hybridized with Tabu Search (MO-VNSTS) "

Truck cost → for all instances, **Cost(GA)= Cost (benchmark) H1 and H2**

Energy cost → for all instances, **Cost(GA)= Cost (benchmark) H1 and H2**

Table 6 : Numerical results on benchmark instances

Instances		Truck cost (F1)			Energy cost (F2)		
Instance	file name	Truck cost (GA)	Truck cost (benchmark H1)	Truck cost(benchmark H2)	Energy cost (GA)	Energy cost (benchmark H1)	Energy cost (benchmark H2)
4	inst_4_1_4.txt	351	351	351	136	136	136
4	inst_4_2_4.txt	1096	1096	1096	136	136	136
4	inst_4_3_4.txt	1542	1542	1542	108	111	111
5	inst_5_1_4.txt	750	1835	1835	199	194	194
5	inst_5_2_4.txt	1835	750	750	183	199	199
5	inst_5_3_4.txt	1923	1923	1923	226	214	214
6	inst_6_1_5.txt	1166	1166	1166	180	192	192
6	inst_6_2_5.txt	1033	1033	1033	192	183	183
6	inst_6_3_5.txt	1440	1440	1440	296	160	160
7	inst_7_1_5.txt	1263	1263	1263	220	282	287
7	inst_7_2_5.txt	1792	1792	1792	253	306	306
7	inst_7_3_5.txt	1797	1797	1797	296	293	293
8	inst_8_1_5.txt	1821	1821	1821	266	273	273
8	inst_8_2_5.txt	2510	2510	2510	276	283	283
8	inst_8_3_5.txt	2520	2520	2520	329	281	281
9	inst_9_1_7.txt	891	891	891	310	357	360,2
9	inst_9_2_7.txt	1650	1650	1650	344	355,8	366
9	inst_9_3_7.txt	2331	2331	2331	350	372,8	374,8
10	inst_10_1_7.txt	1588	1588	1588	345	431	436
10	inst_10_2_7.txt	2098	2098	2098	377	392	394,2
10	inst_10_3_7.txt	3102	3102	3102	394	422,6	425,6
11	inst_11_1_7.txt	1276	1276	1276	319	406	406
11	inst_11_2_7.txt	1876	1876	1876	382	400	406,2
11	inst_11_3_7.txt	2042	3699	3699	401	495	498,6
12	inst_12_1_7.txt	2035	2035	2035	412	494	495
12	inst_12_2_7.txt	2688	2688	2688	445	499	499
12	inst_12_3_7.txt	4387	3699	3699	607	495	498,6
20	inst_20_10_15.txt	9595	7338	7338	1070	801,2	776
20	inst_20_7_15.txt	5001	5001	5001	760	772,6	750
30	inst_30_10_20.txt	6401	6401	6401	1209	1284,8	1201
30	inst_30_15_20.txt	9529	9529	9529	1104	1220,4	1182
30	inst_30_7_20.txt	5081	5081	5081	1250	1279,8	1216

GAP   Total cost   Comparison with benchmark   **Bing instances**   Computational Time   Model's Limits

Table 5 : Exact method results for big instances

File Name	Lower bound	Status	MipGap	Computational Time(s)
inst_11_3_7.txt	1219,5	TimeLimit	0.0340303403	7200,031
inst_12_3_7.txt	2063,5	TimeLimit	0.0193845408	7200,029
inst_20_7_15.txt	2865	TimeLimit	0.1178010471	7200,034
inst_20_10_15.txt	4027	TimeLimit	0.0737521728	7200,026
inst_30_7_20.txt	3108,5	TimeLimit	0.2115168088	7200,082
inst_30_10_20.txt		**Infeasible**		**0,141**
inst_30_15_20.txt	5316,5	TimeLimit	0.0958337252	7200,032

Table 6 : Approximative method results for big instances

File Name	Approximative cost	Computational Time(s)	Population size	Generations
inst_11_3_7.txt	1221,5	73,379	50	100
inst_12_3_7.txt	2497	69,433	50	100
inst_20_7_15.txt	2880,5	124,687	50	100
inst_20_10_15.txt	5332,5	130,171	50	100
inst_30_7_20.txt	3165,5	180,866	50	100
inst_30_10_20.txt	**3805**	**179,821**	**50**	**100**
inst_30_15_20.txt	5316,5	191,094	50	100

● Instance_30_10_20 : **Infeasible** for Gurobi, **GA** found **approximative** total cost = 3805

- Instances (**inst_11_3_7, inst_12_3_7, inst_20_7_15, inst_20_10_15, inst_30_7_20, inst_30_15_20**): **Reached time limit = 7200 (2h)** for Gurobi
Gurobi found **lower bound** (theorical value with constraints relaxation**) < GA approximative values for the instances**

GAP   Total cost   Comparison with benchmark   **Bing instances**   **Computational Time**   **Model's Limits**

Motivation and context   Mathematical formula   Experimental study   • Benchmark   • Comparison
Genetic Algorithm (GA)   Genetic operators   Conclusion and discussion   • Computational setup
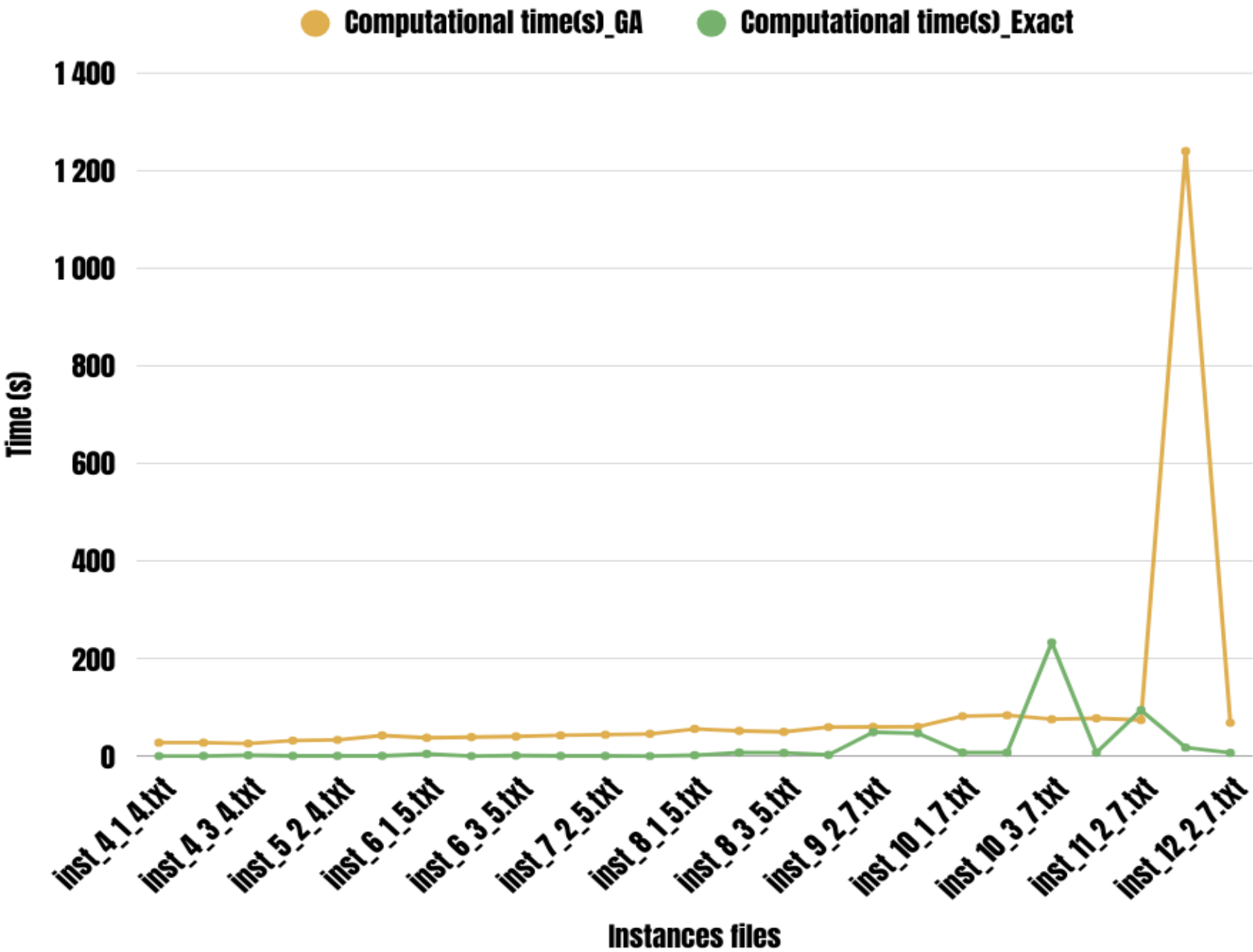
GAP   Total cost   Comparison with benchmark   Bing instances   Computational Time   Model's Limits

Three distinct difficulty regions:

📍 Eazy region

[**4** to **7**] trucks and [**5** to **10**] containers, the solver found optimal solutions easily

📍 Transition region

Starting from **7** trucks and **10** to **12** containers, approximately **33%** of instances reach the time limit; the model is still solvable but demands more computational power.
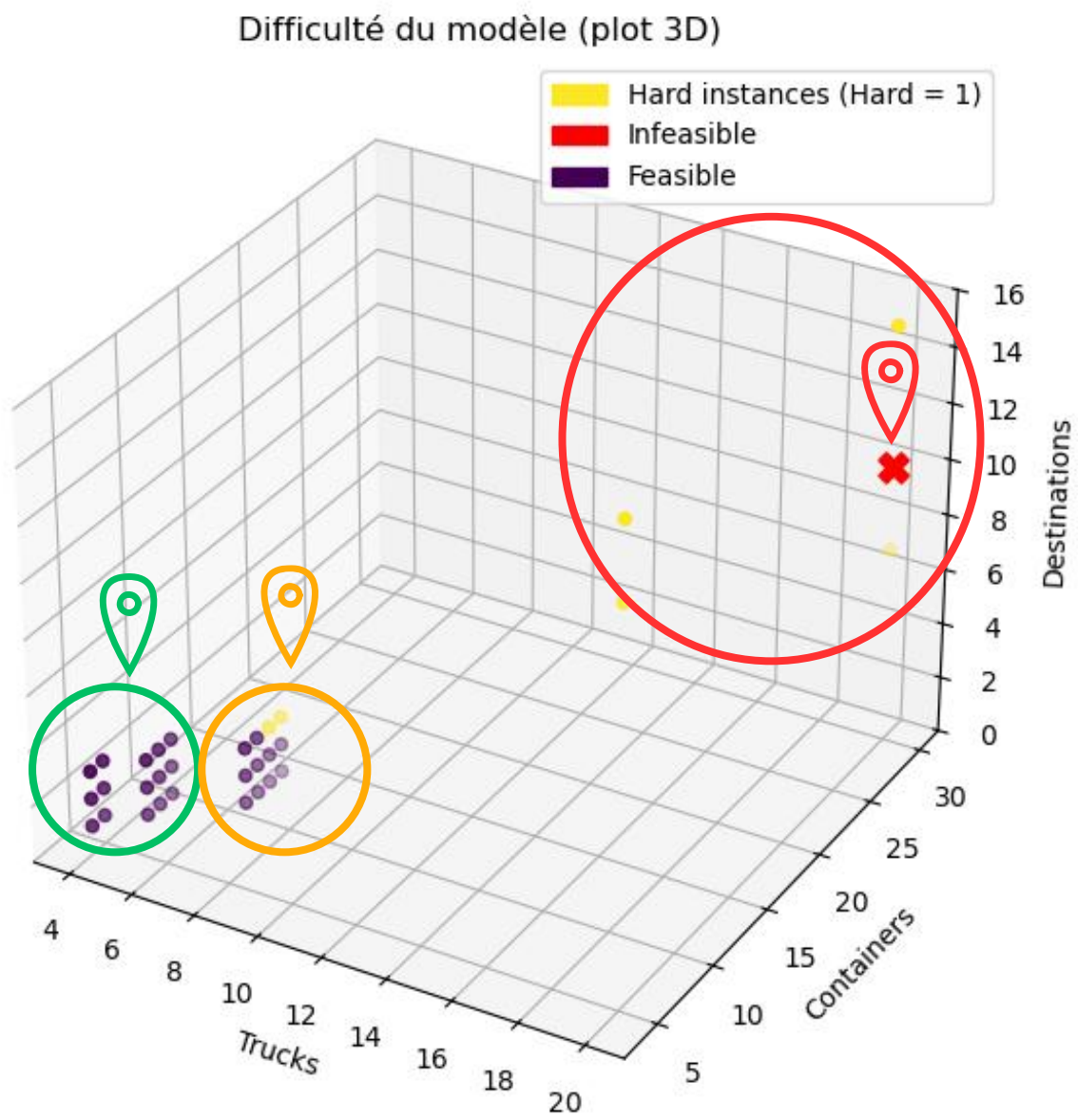
📍 Difficult region (large instances)

15 trucks, **20** trucks, and **7-10** destinations; all instances reached the time limit

20 trucks, **30** containers, **7-15** destinations; **67%** hit time limit.

20 trucks, **30** containers, and **10** destinations; the instance is unfeasible.

Difficulté du modèle (plot 3D)

Legend:
- Hard instances (Hard = 1)
- Infeasible
- Feasible

Conclusion, improvements & future work

Big instances->GA faster-> Lose precision

Big instances->Exact_methode found optimal-> bigger time execution

Initial initialization plays a big role on the quality of the GA solutions

Solution representation plays a big role in the effectiveness of the genetic algorithm

Tradeoff (Optimal solutions, and Running time)

The results are still being stabilized, but the observed trends indicate that GA can provide a quick alternative, especially for large instances.

- **Experiment with other genetic operations, fitness sharing,random immigration, other combinations of crossover, ...**
- **Move to Multi-Objective GA (NSGA-II)(Use NSGA-II to generate a Pareto front of best compromises)**
- **Parameter tuning: population size, crossover rate, mutation rate, elitism.**
- **Explore hybridized metaheuristics**

References

[1] : Tran-Dang et al , (2021)
[2] : Chargui et al , (2019)

Thank you