Algorithms for online decision making in OM

Willem van Jaarsveld - w.l.v.jaarsveld@tue.nl
Eindhoven University of Technology

November 22, 2023

Outline

Canonical approach to decision optimization in operations management (OM)

Examples of impactfull applications

Outline

Canonical approach to decision optimization in operations management (OM)

Examples of impactfull applications

$\Rightarrow$ Data-driven Decision Making

Outline

Canonical approach to decision optimization in operations management (OM)

Examples of impactfull applications

$\Rightarrow$ Data-driven Decision Making

- Deep Reinforcement Learning

Content/acknowledgement

Published:

Deep reinforcement learning for inventory control: A roadmap (with Robert Boute, Joren Gijsbrechts, Nathalie VanVuchelen),

Policies for the Dynamic Traveling Maintainer Problem with Alerts (with Paulo Oliveira da Costa, Peter Verleijsdonk, Simon Voorberg, Alp Akçay, Stella Kapodistria, Yingqian Zhang)

Ongoing:

Scalable policies for the dynamic traveling multi-maintainer problem with alerts (with Peter Verleijsdonk, Stella Kapodistria)

Deep Controlled Learning for Inventory Control (with Remco Dijkman, Tarkan Temizöz, Christina Imdahl, Douniel Lamghari-Idrissi)

Value-based versus Policy-based Reinforcement Learning for Dynamic Vehicle Routing Problems (with Fabian Akkerman and Martijn Mes)

Canonical approach

- What is my objective?
- What are my decision variables?
- What are my constraints?
  - chance constraints, robust constraints, etc.

Canonical approach (2)

$\Rightarrow$ The “puzzle” paradigm

Canonical approach (2)

$\Rightarrow$ The "puzzle" paradigm

$$\min_{\vec{x}} c(\vec{x})$$
$$\text{s.t. } \vec{x} \in X$$

Canonical approach (2)

⇒ The "puzzle" paradigm

$$\min_{\vec{x}} c(\vec{x})$$
$$\text{s.t. } \vec{x} \in X$$

Theory: Needle-in-the-Haystack - NP-hard / NP-complete.

Canonical approach (2)

$\Rightarrow$ The "puzzle" paradigm

$$\min_{\vec{x}} c(\vec{x})$$
$$\text{s.t. } \vec{x} \in X$$

Theory: Needle-in-the-Haystack - NP-hard / NP-complete.
Practice: Mixed-integer Programming / ALNS

Canonical approach (2)

$\Rightarrow$ The "puzzle" paradigm

$$\min_{\vec{x}} c(\vec{x})$$
$$\text{s.t. } \vec{x} \in X$$

Theory: Needle-in-the-Haystack - NP-hard / NP-complete.
Practice: Mixed-integer Programming / ALNS

Underlies most optimization software nowadays:

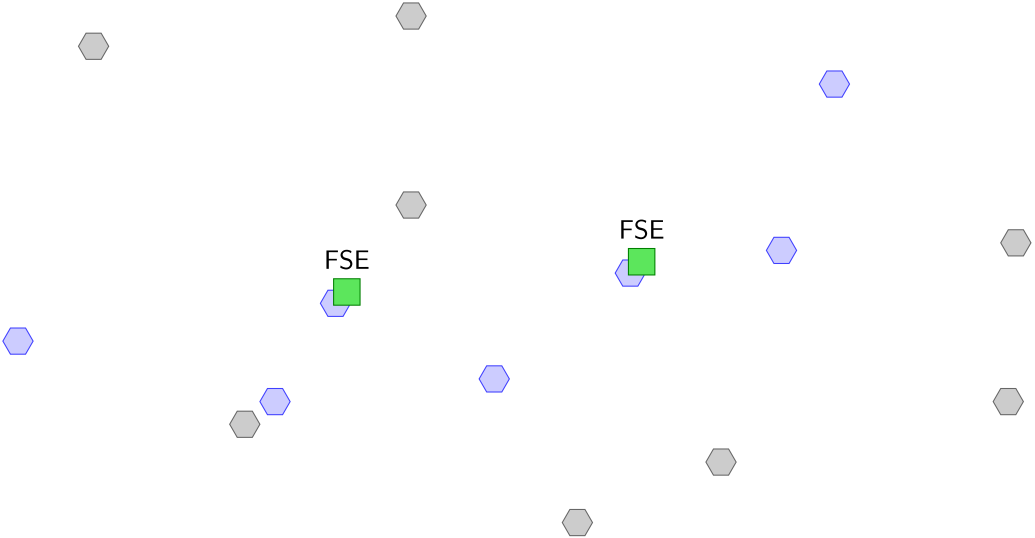
- Flexible
- Well-studied, tried, and trusted
- Joint optimization of decisions x_i that constitute $\vec{x} = \{x_1, x_2, \dots, x_n\}$.

Example (with Philips)

Goal: On-time equipment maintenance

- Field service engineers ( )
- Hospital/equipment location ( )
- Equipment due for preventive maintenance ( )

Example (with Philips)

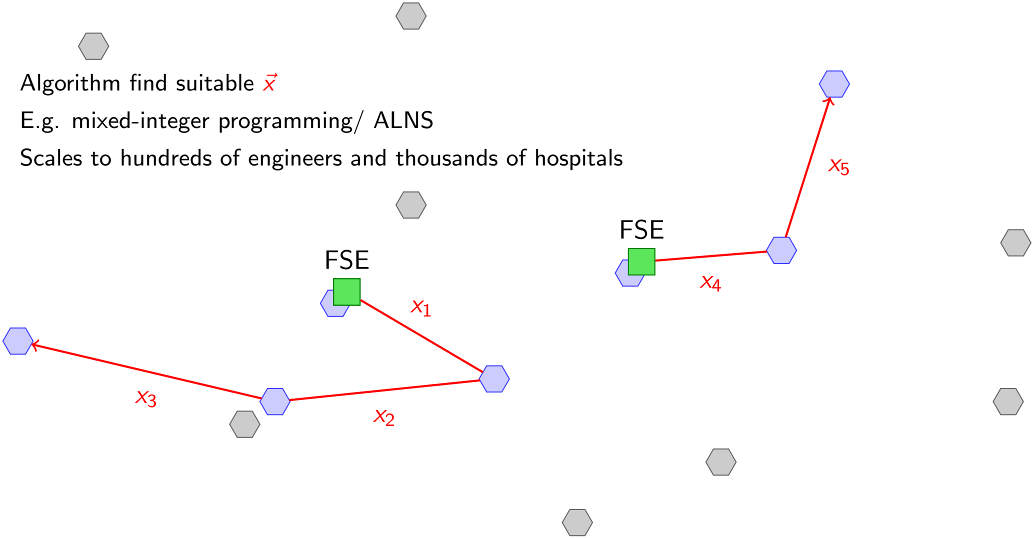


Example (with Philips)

Algorithm find suitable $\vec{x}$

E.g. mixed-integer programming/ ALNS

Scales to hundreds of engineers and thousands of hospitals



Example: Tactical Spare Parts Inventory Planning

Stocking policy for spares $i \in \{1, \ldots, N\}$ to achieve overall weighted fill rate target at minimum holding costs

Example: Tactical Spare Parts Inventory Planning

Stocking policy for spares $i \in \{1, \ldots, N\}$ to achieve overall weighted fill rate target at minimum holding costs:

$$\min_{\vec{x}=\{S_1,S_2,\ldots,S_n\}} \sum_{i=1}^{N} h_i S_i$$

$$\text{s.t.} \sum_{i \in I} w_i F_i(S_i) \geq 95\%$$

$S_i \in \{0, 1, 2 \ldots\}$: Base-stock level for SKU i

$F_i(S_i)$: Long-term fill-rate when using base-stock S_i for SKU i

Outline

Canonical approach to decision optimization in operations management (OM)

Examples of impactfull applications

$\Rightarrow$ Data-driven Decision Making

Example (with Philips)

Goal: On-time equipment maintenance

- Field service engineers ( )
- Hospital/equipment location ( )
- Equipment due for preventive maintenance ( )

Example (with Philips)

Goal: On-time equipment maintenance

- Field service engineers ()
- Hospital/equipment location ()
- Equipment due for preventive maintenance ()

Opportunity: Predict maintenance using equipment IoT data
Maintain only when needed : data-driven maintenance!

Example (with Philips)

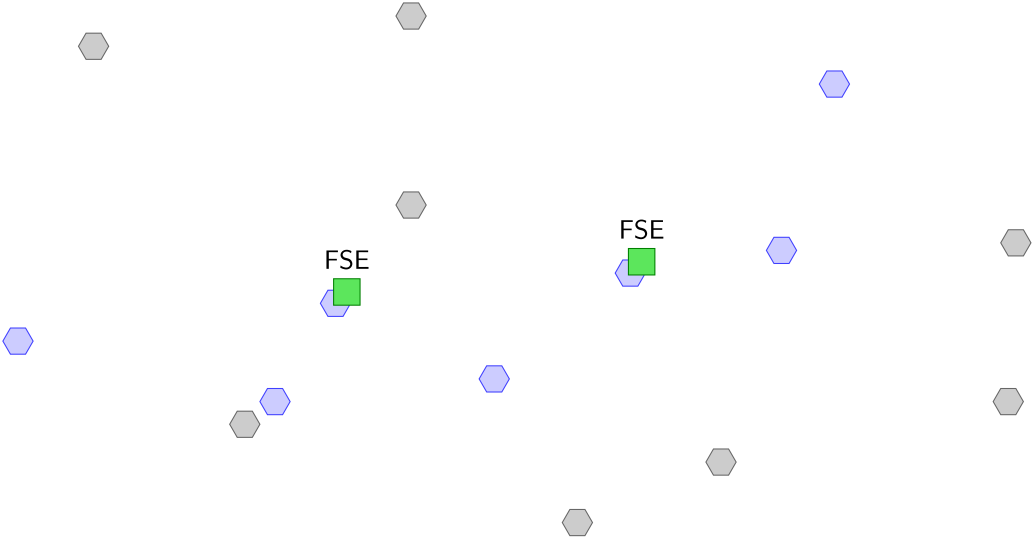
Goal: On-time equipment maintenance

- Field service engineers ()
- Hospital/equipment location ()
- Equipment due for preventive maintenance ()

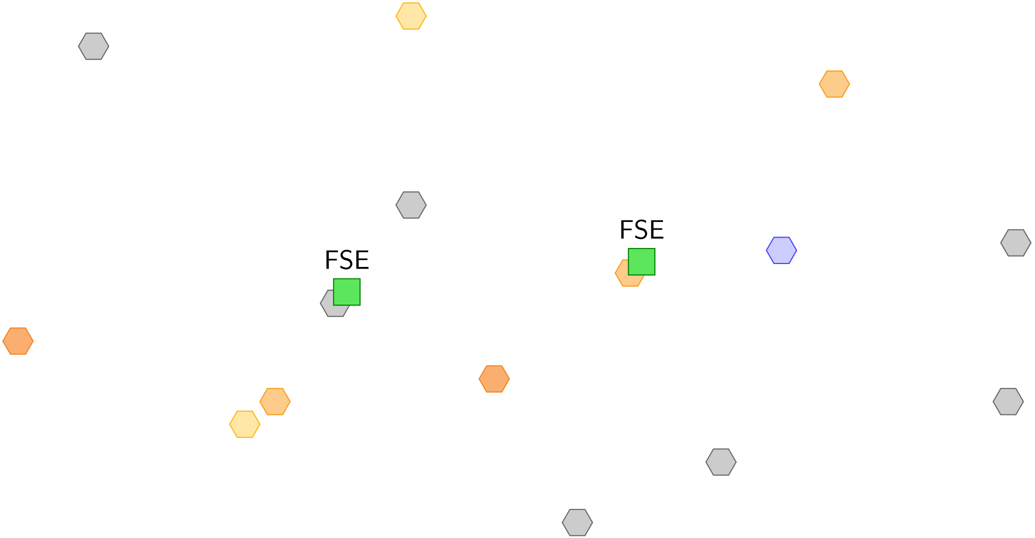
Opportunity: Predict maintenance using equipment IoT data
Maintain only when needed : data-driven maintenance!

- Equipment with level 1/2/3 **alert** (//)
- Equipment with diminished capability ()

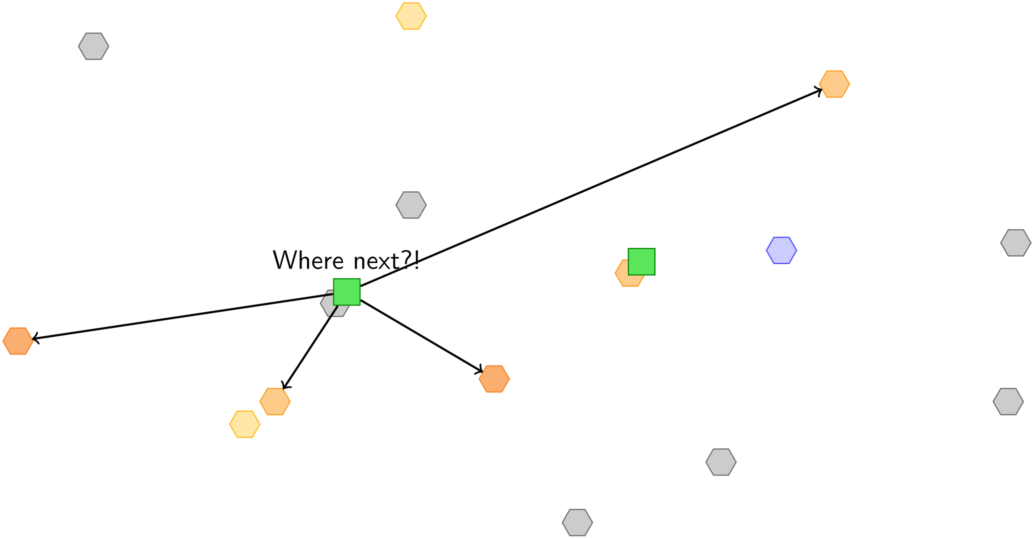
Example (with Philips)



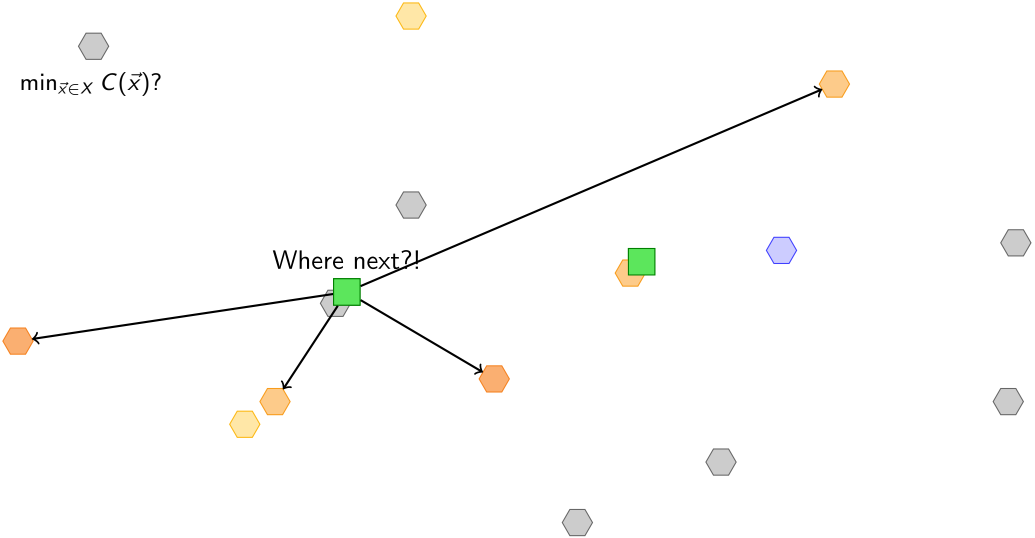
Example (with Philips)



Example (with Philips)

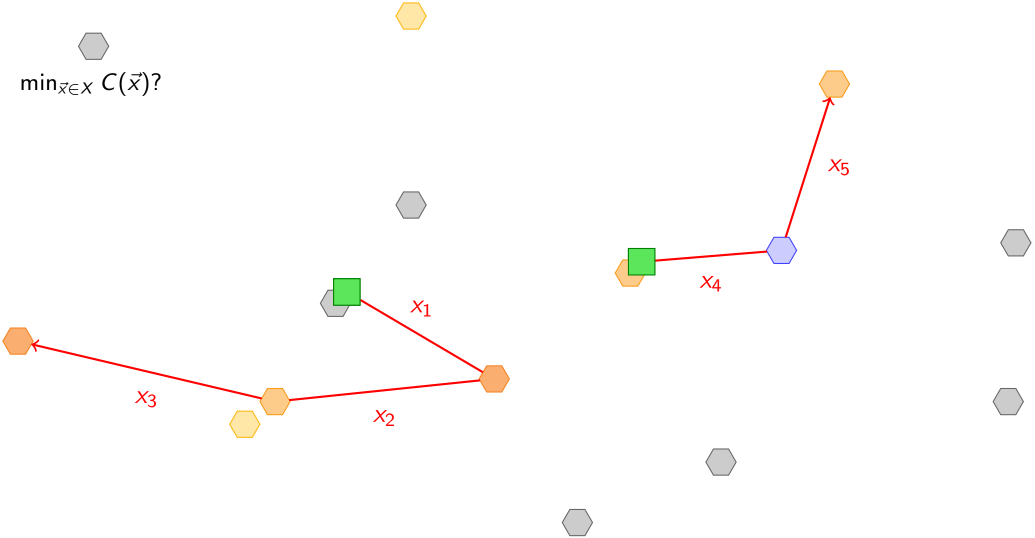


Example (with Philips)



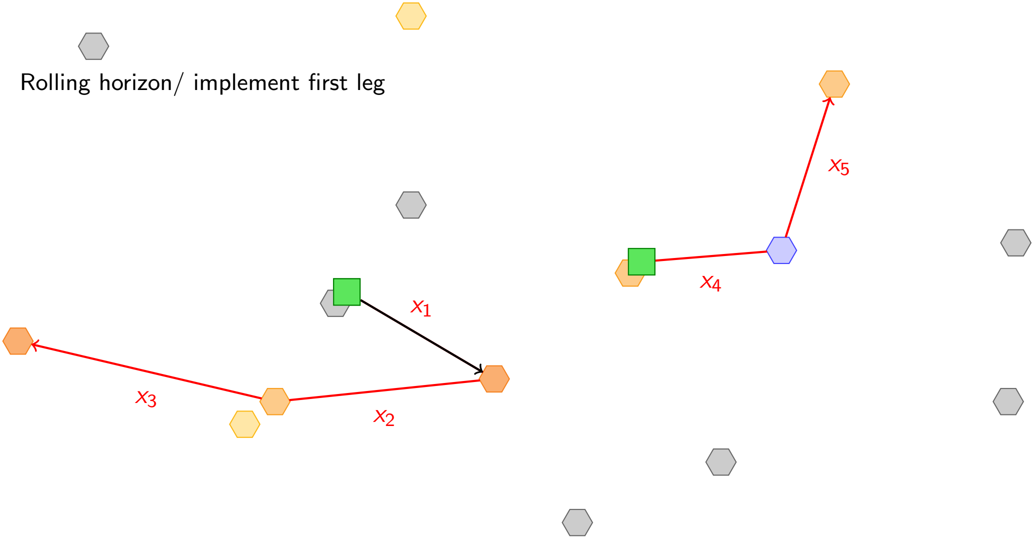
Example (with Philips)

$\min_{\vec{x} \in X} C(\vec{x})$?



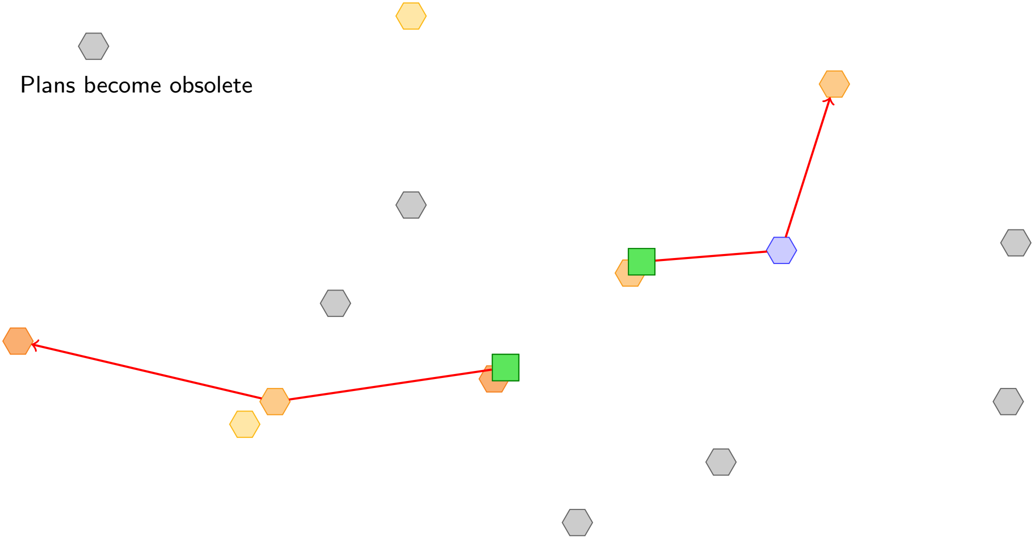
Example (with Philips)

Rolling horizon/ implement first leg

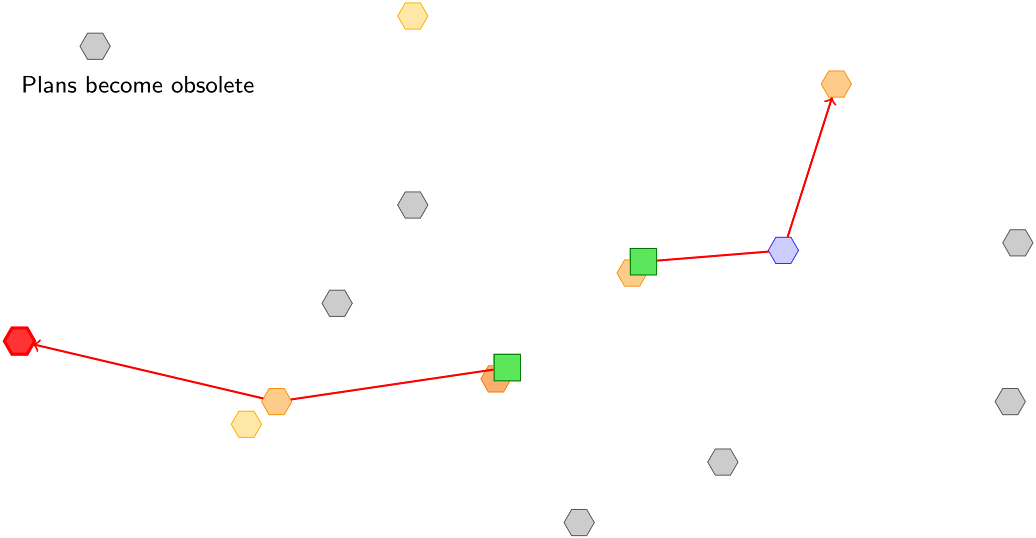


Example (with Philips)

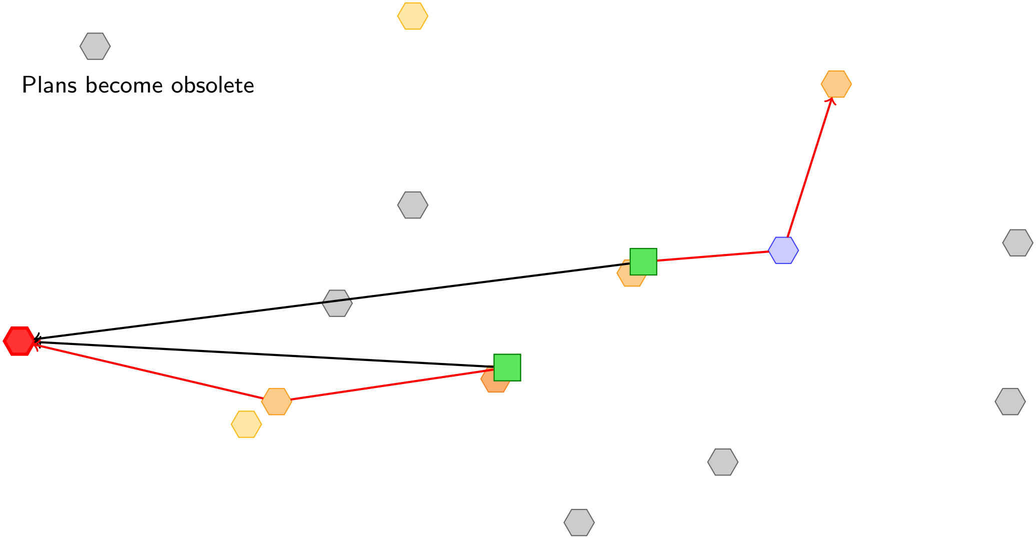
Plans become obsolete



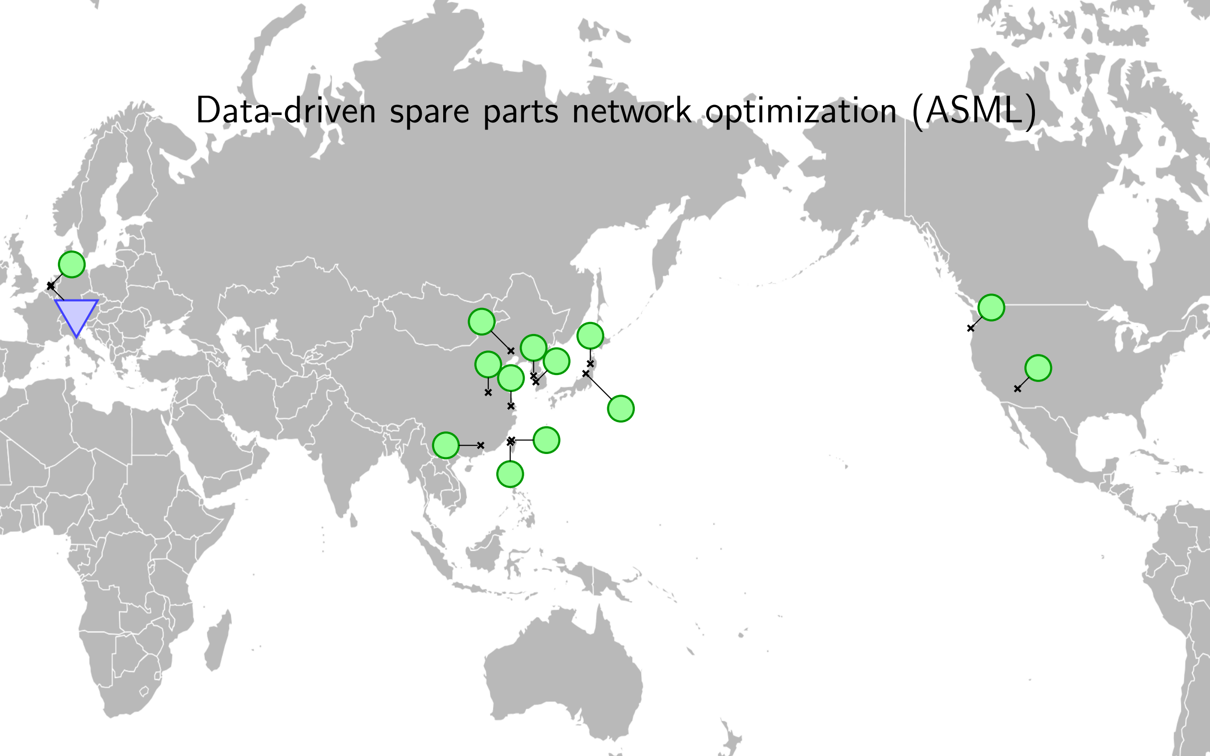
Example (with Philips)

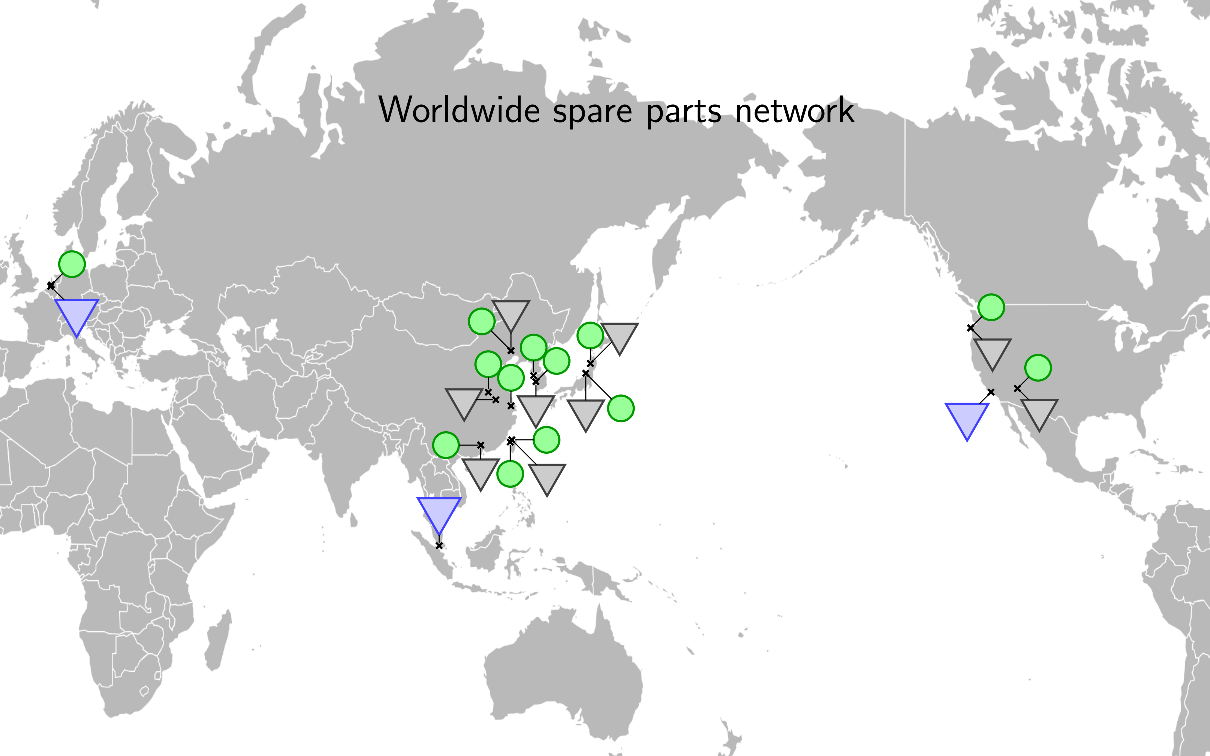


Example (with Philips)

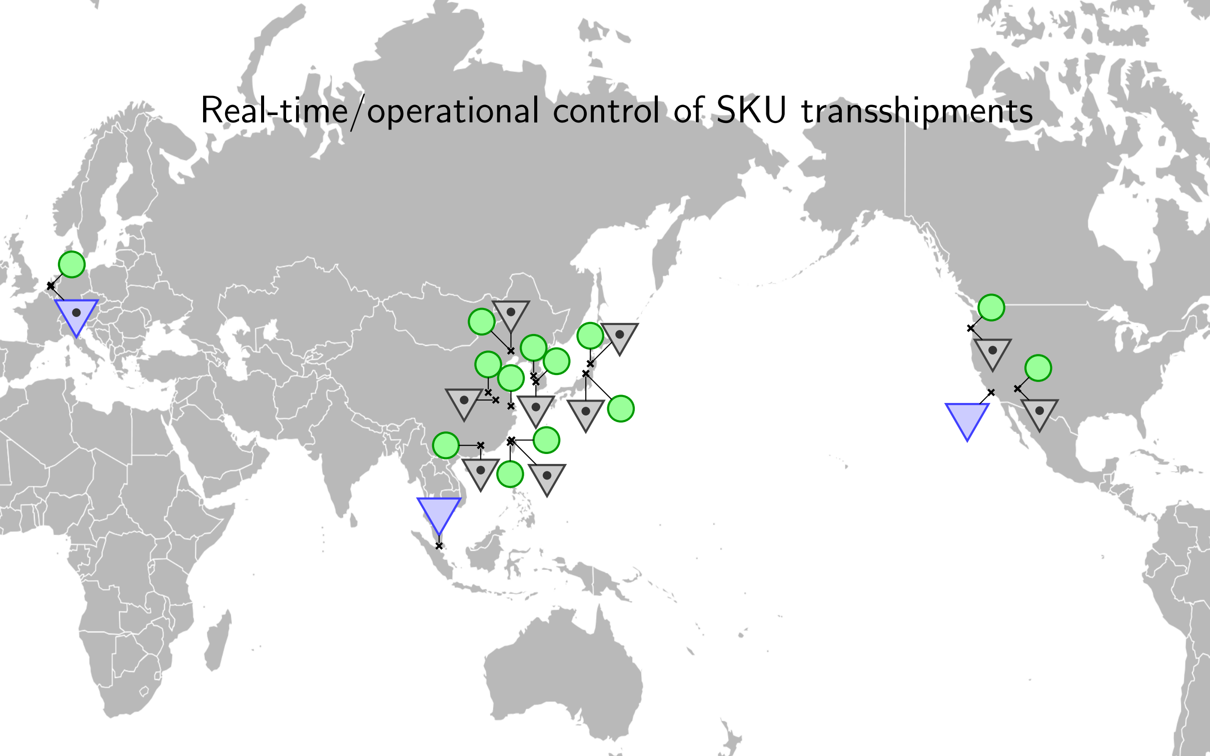


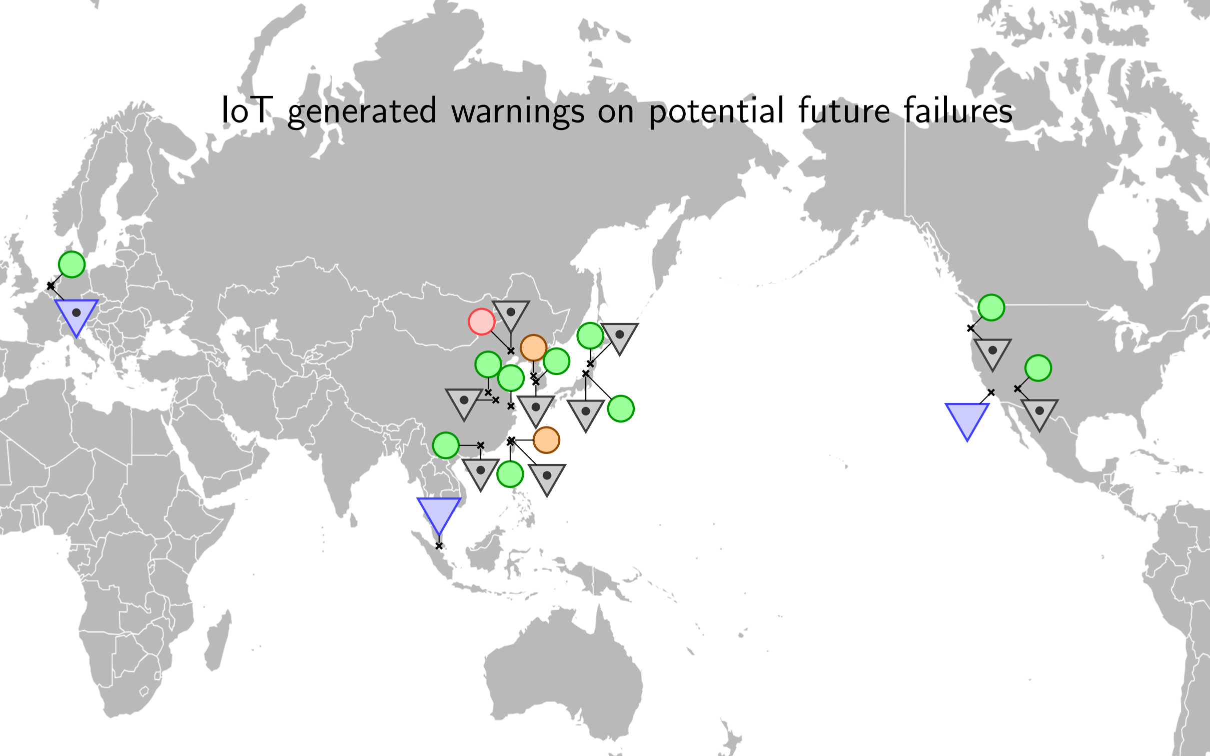
Data-driven spare parts network optimization (ASML)



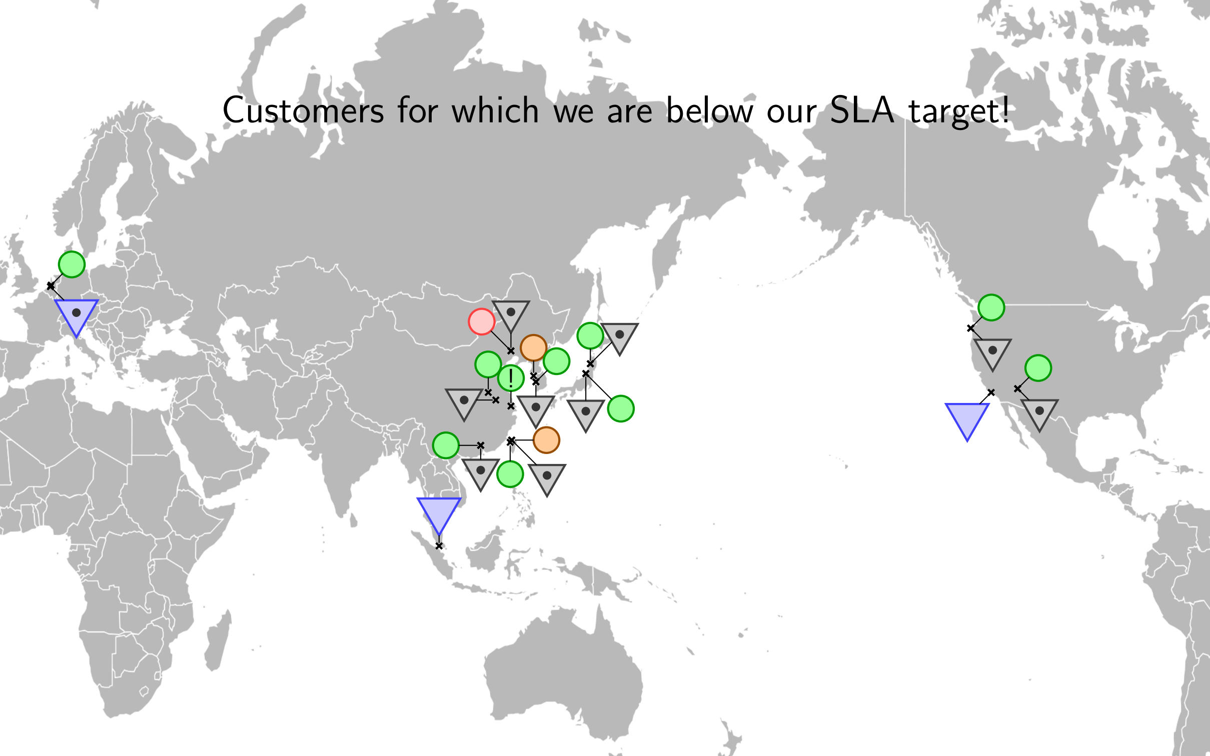

Worldwide spare parts network

Real-time/operational control of SKU transshipments

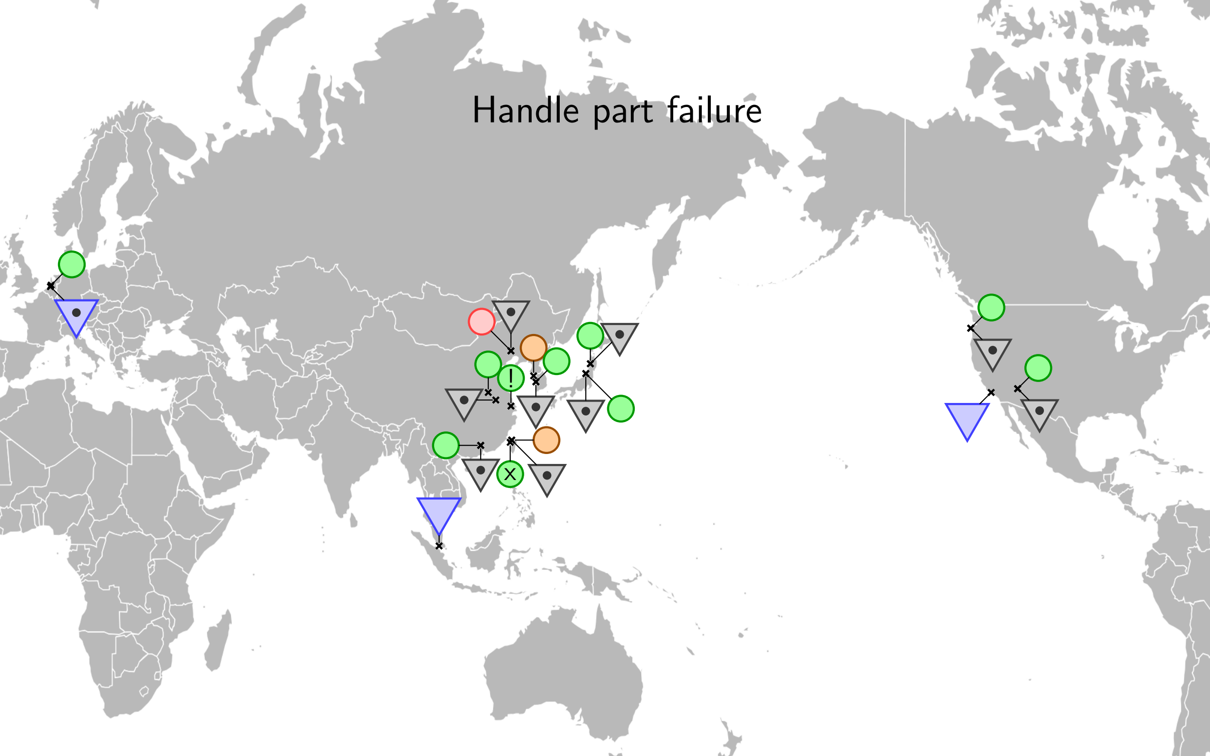



IoT generated warnings on potential future failures

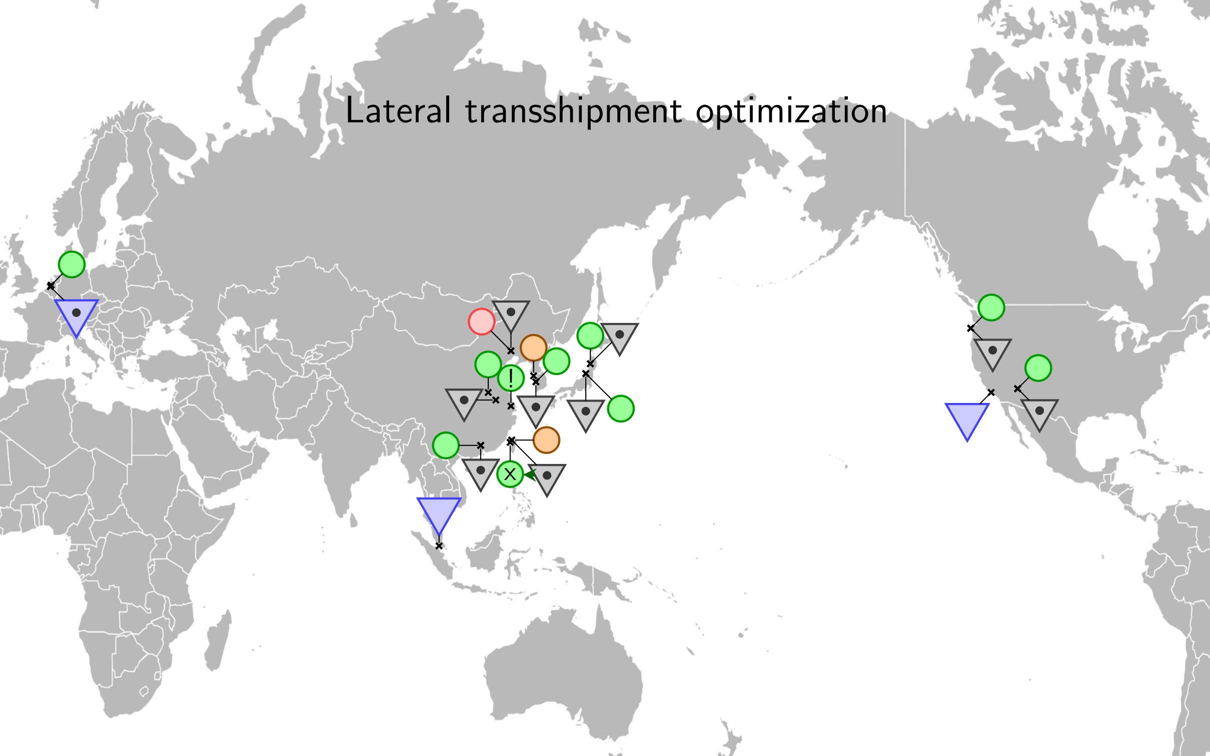
Customers for which we are below our SLA target!



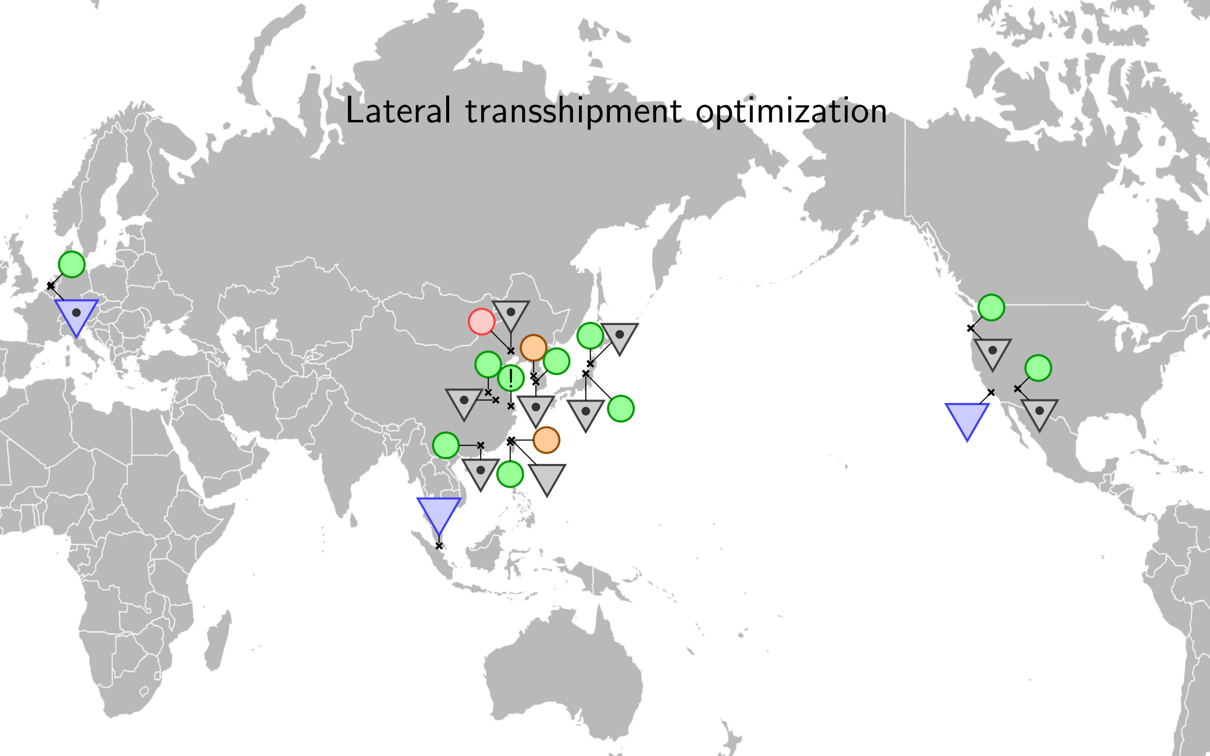
Handle part failure



Lateral transshipment optimization

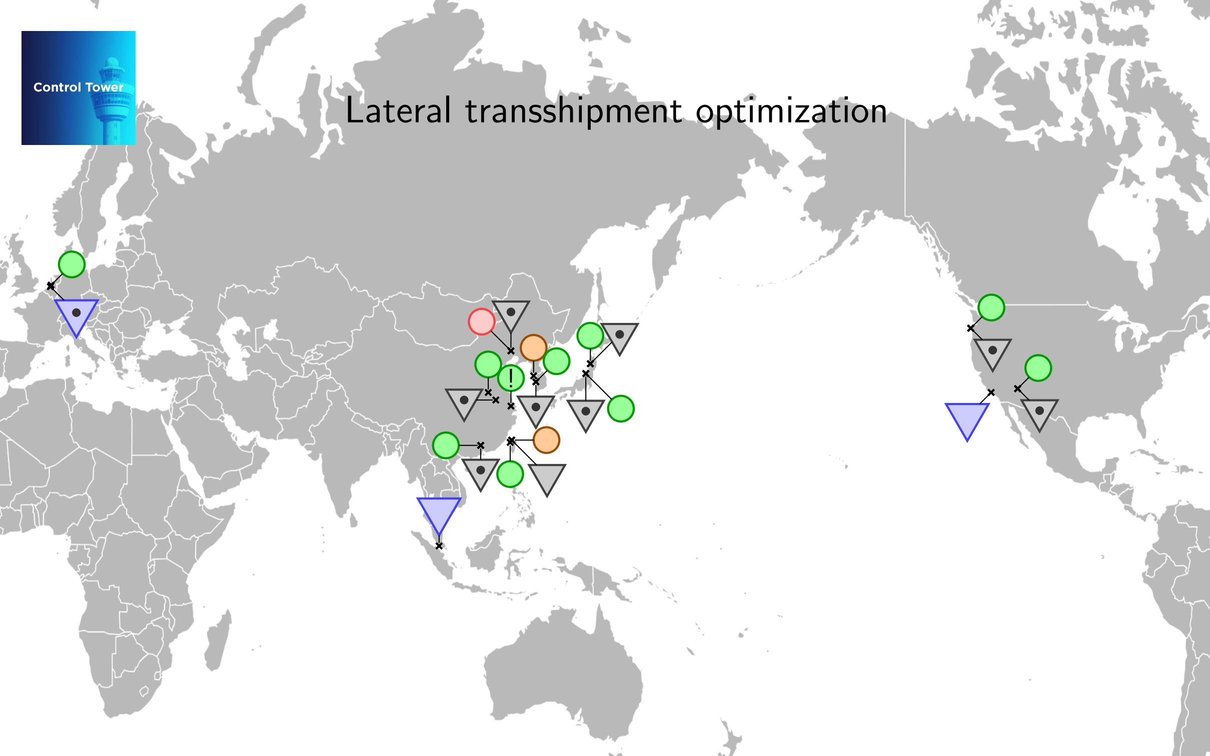


Lateral transshipment optimization



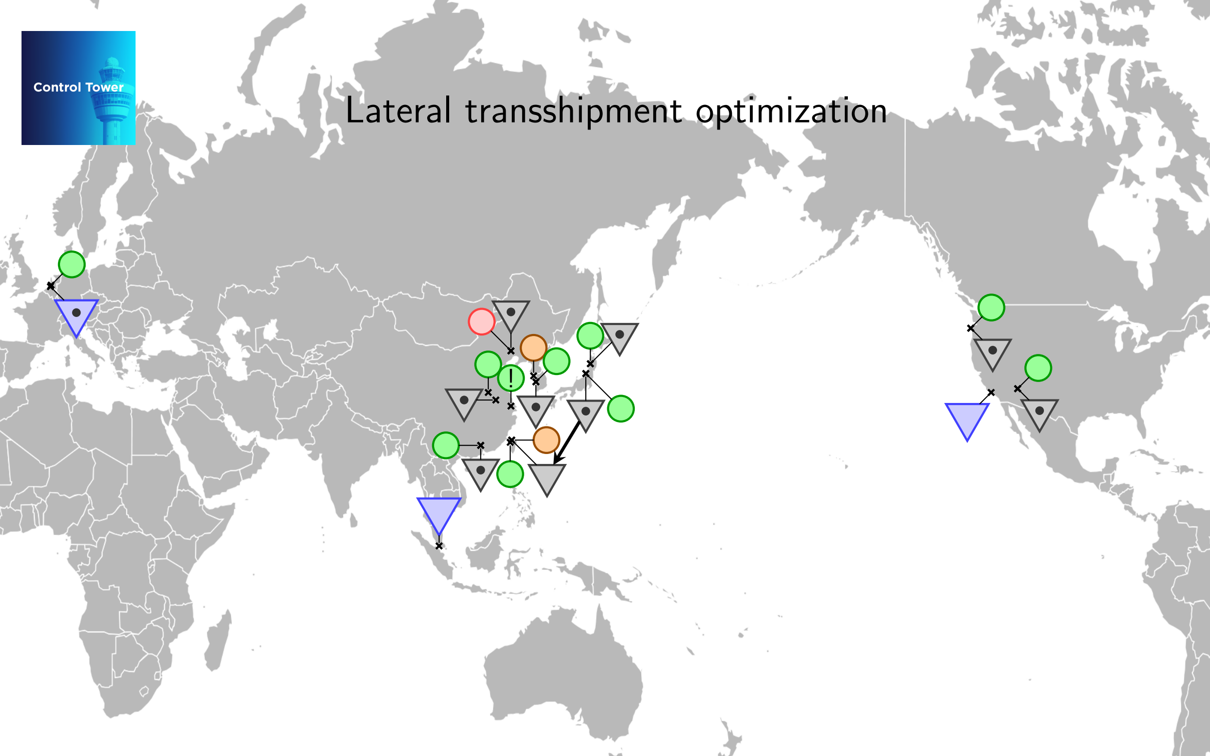
Control Tower

Lateral transshipment optimization

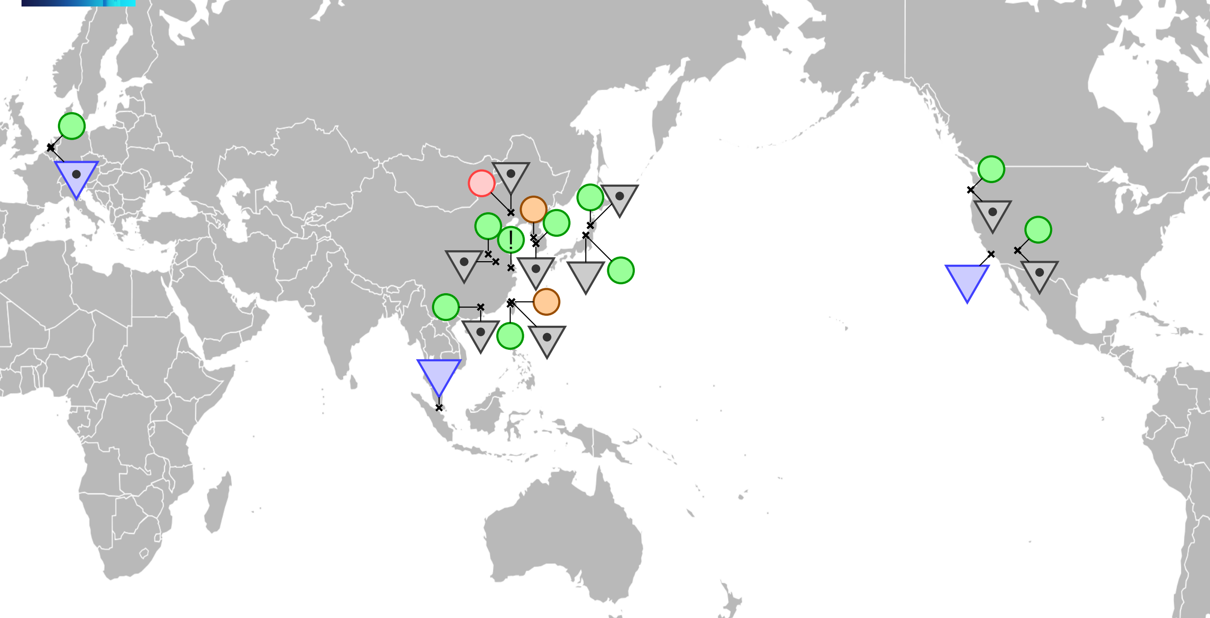


Control Tower

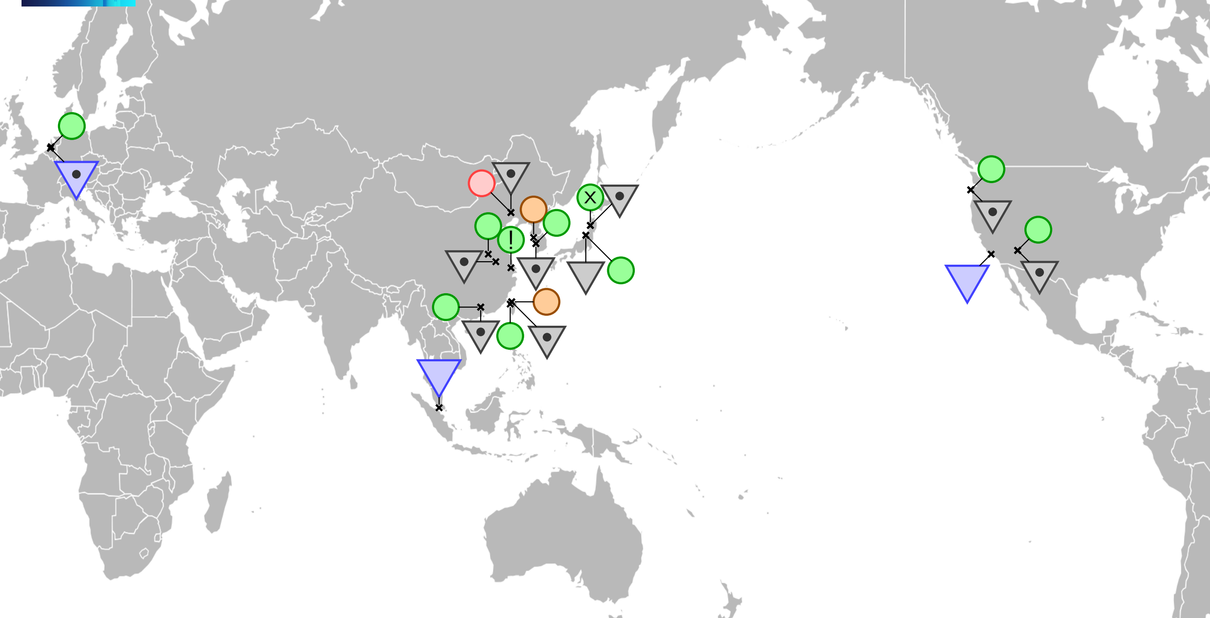
Lateral transshipment optimization



Lateral transshipment optimization

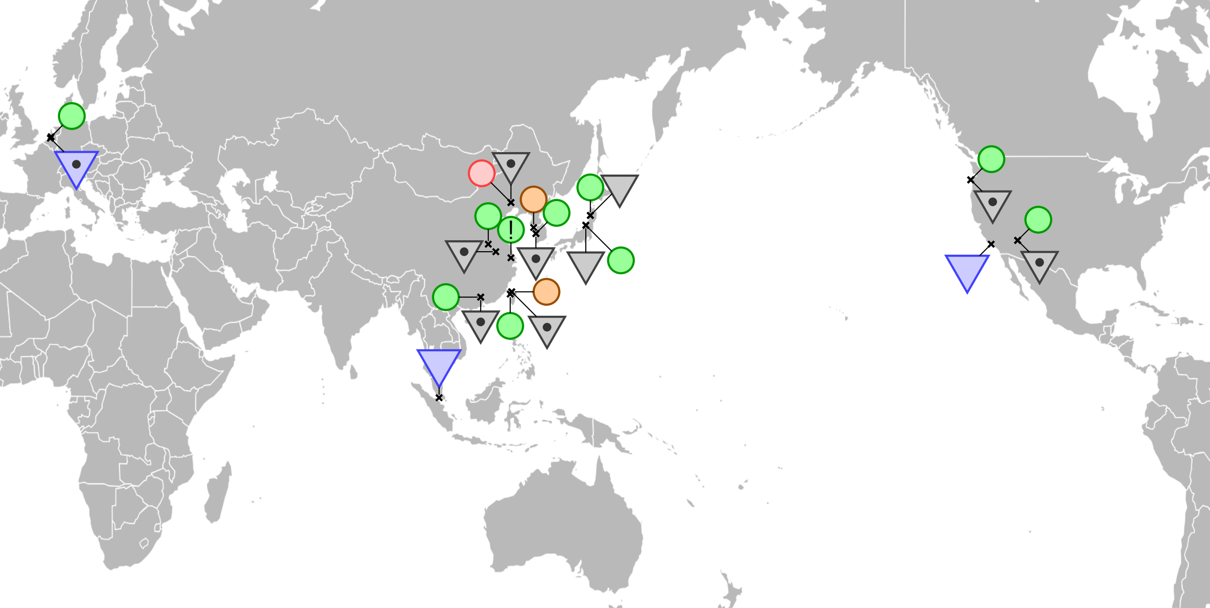


Lateral transshipment optimization

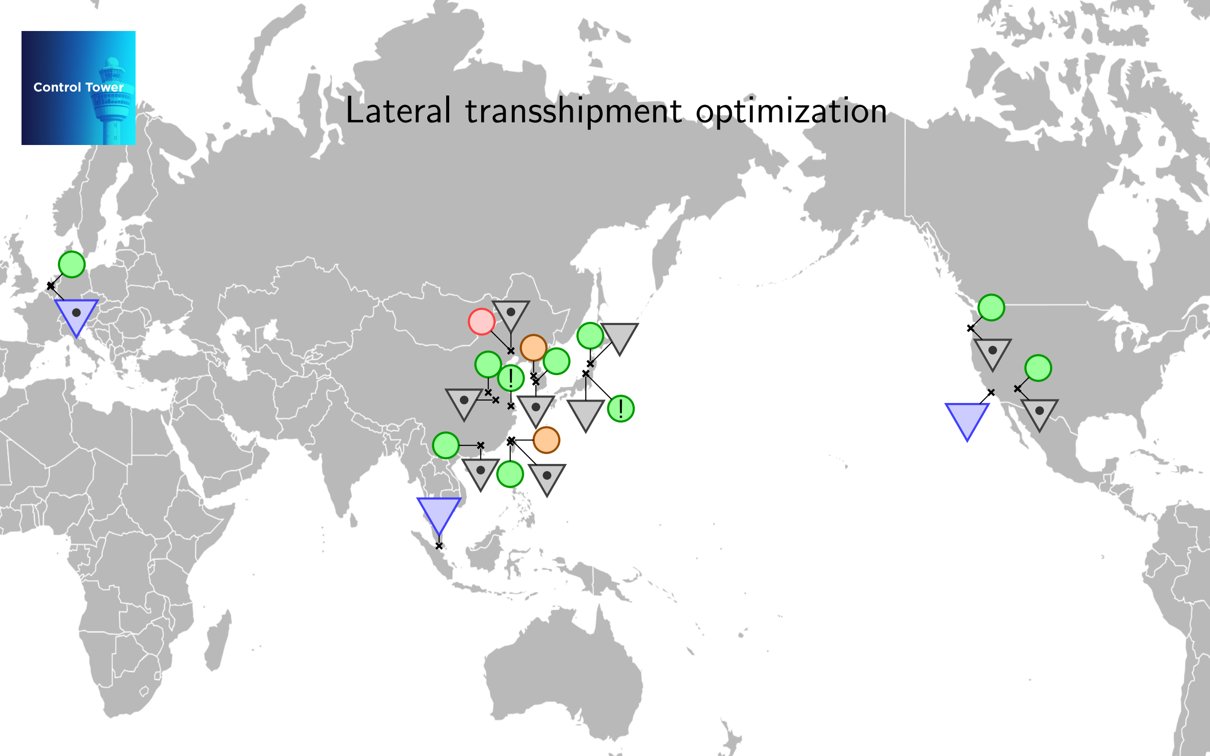


Control Tower

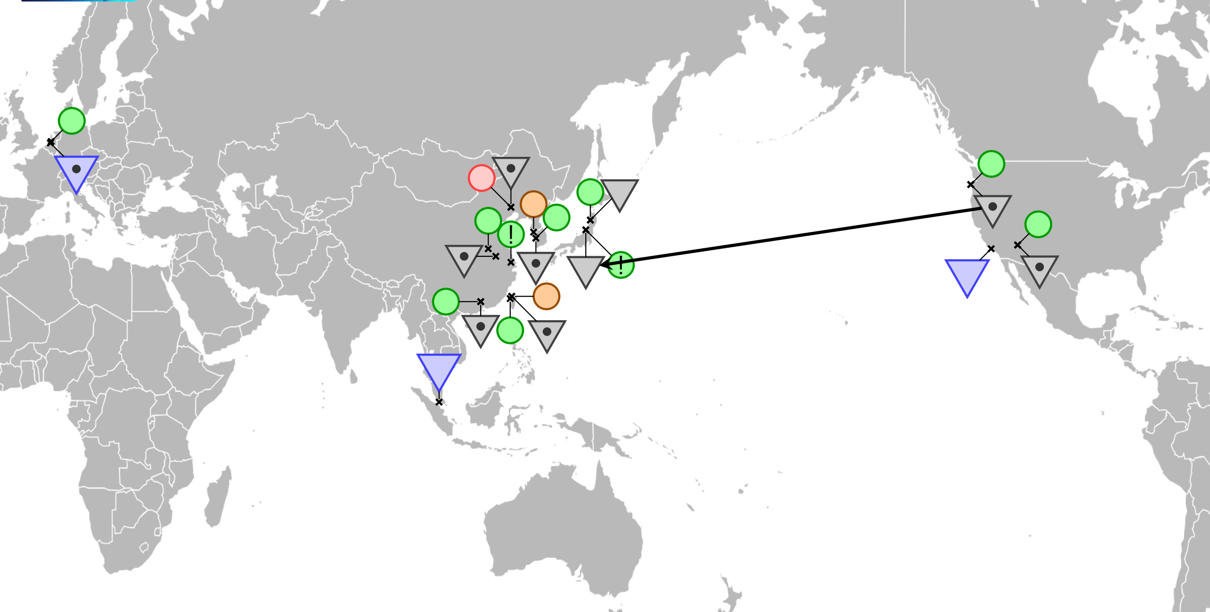
Lateral transshipment optimization



Lateral transshipment optimization

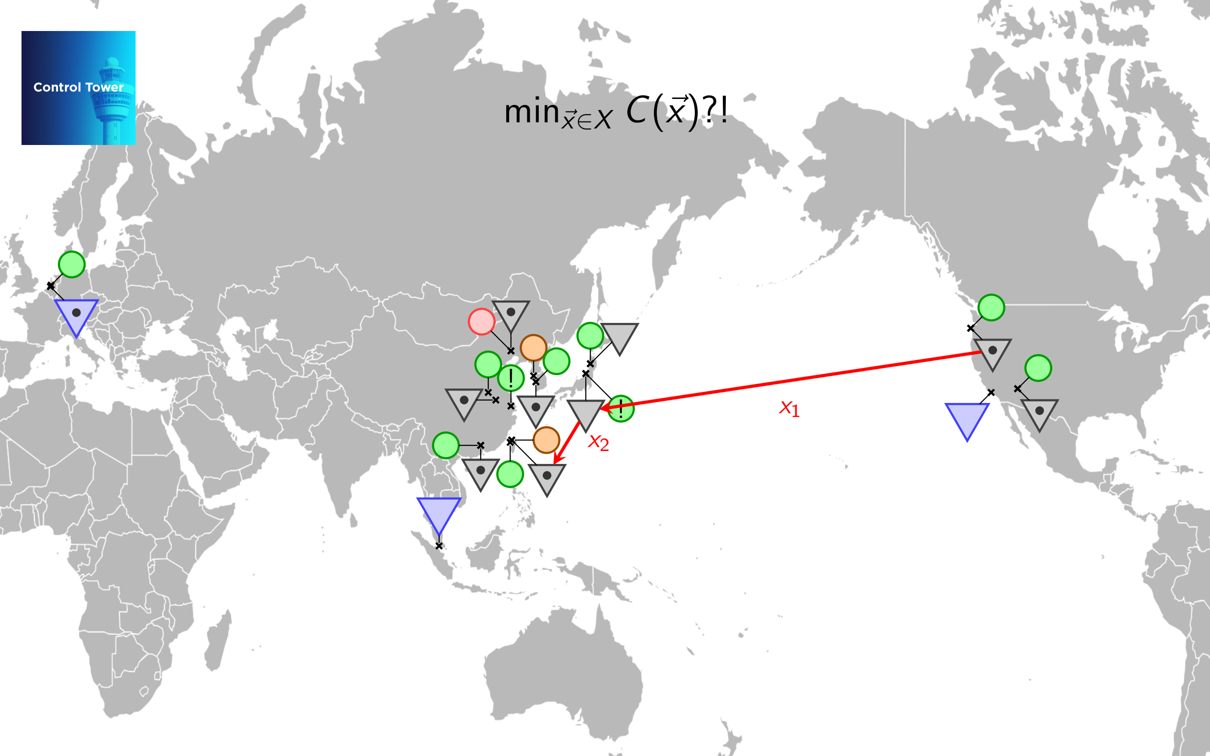


Lateral transshipment optimization



Control Tower

$$\min_{\vec{x} \in X} C(\vec{x})?!$$



Recap and outlook

$$\min_{\vec{x} \in X} c(\vec{x})$$

The "puzzle" paradigm to optimization:

- Flexible
- Well-studied, tried, and trusted
- Joint optimization of decisions x_i that constitute $\vec{x} = \{x_1, x_2, \ldots, x_n\}$.

Recap and outlook

$$\min_{\vec{x} \in X} c(\vec{x})$$

The "puzzle" paradigm to optimization:

- Flexible
- Well-studied, tried, and trusted
- Joint optimization of decisions x_i that constitute $\vec{x} = \{x_1, x_2, \ldots, x_n\}$.
- Not a natural model for data-driven/online optimization

Recap and outlook

$$\min_{\vec{x} \in X} c(\vec{x})$$

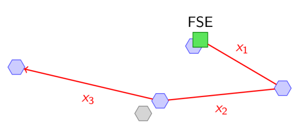
The "puzzle" paradigm to optimization:

- Flexible
- Well-studied, tried, and trusted
- Joint optimization of decisions x_i that constitute $\vec{x} = \{x_1, x_2, \ldots, x_n\}$.
- Not a natural model for data-driven/online optimization

Observation: Data-driven/online optimization is like a single-player "game", where alternating decisions and random events determine the eventual outcome.

Puzzles and Games

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

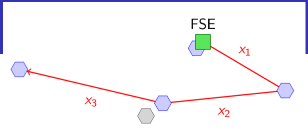


Static "puzzles" are often NP-complete:

- Difficult to find a solution
- Validity and objective value of solution can be verified in polytime.

Puzzles and Games

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

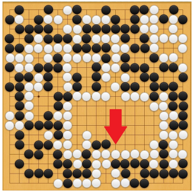


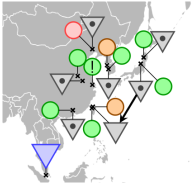
Static "puzzles" are often NP-complete:

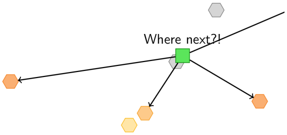
- Difficult to find a solution
- Validity and objective value of solution can be verified in polytime.

Dynamic "games" are often PSPACE-complete:

- Difficult to find a move that eventually enables a favorable outcome
- Evaluating the quality of a move is also difficult





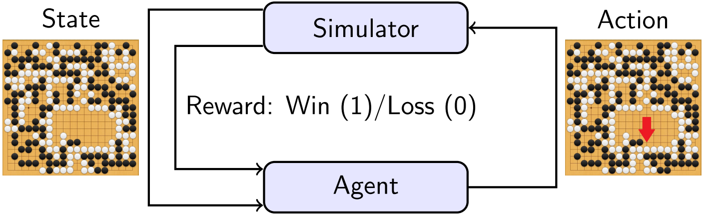



Google Deep
Challenge Ma
8 - 15 March 20
ALPHAGO
00:05:30
LEE SEDOL
00:28:28
AlphaGo
Lee Sedol

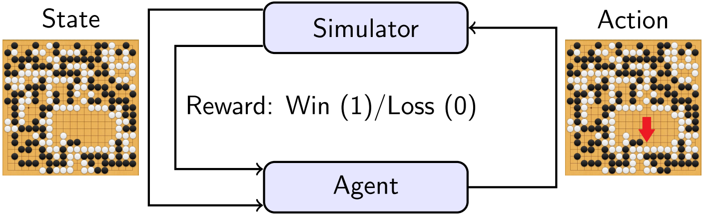
DRL powered software agent AlphaGo beats
top player Lee Sedol (Silver et al, 2016)

AlphaZero taught itself from scratch
how to master Chess, Go & Shogi (2018)

Deep Reinforced Learning



Deep Reinforced Learning



Interesting:

- Learns solely from data
- Uses a neural network to predict good moves for board positions
- Single algorithm competitive for three PSPACE games
- Math underlying this is $\sim$ Markov Decision Process
    - **Natural model for data-driven/online optimization problems**

AlphaZero/DRL: Anything new here?

"Single Deep Reinforcement Learning algorithm employs distributed computing to achieve superhuman performance in Go, Chess and Shogi."

AlphaZero/DRL: Anything new here?

"Single Deep Reinforcement Learning algorithm employs distributed computing to achieve superhuman performance in Go, Chess and Shogi."

- Reinforcement Learning: Machine learning (regression) which fits a model on data obtained from a simulated environment - Existing field built upon earlier work in approximate dynamic programming & optimal control.

AlphaZero/DRL: Anything new here?

"Single Deep Reinforcement Learning algorithm employs distributed computing to achieve superhuman performance in Go, Chess and Shogi."

- Reinforcement Learning: Machine learning (regression) which fits a model on data obtained from a simulated environment - Existing field built upon earlier work in approximate dynamic programming & optimal control.
- Deep RL: using *deep* neural networks in ADP/RL/optimal control - new

AlphaZero/DRL: Anything new here?

"Single Deep Reinforcement Learning algorithm employs distributed computing to achieve superhuman performance in Go, Chess and Shogi."

- Reinforcement Learning: Machine learning (regression) which fits a model on data obtained from a simulated environment - Existing field built upon earlier work in approximate dynamic programming & optimal control.
- Deep RL: using *deep* neural networks in ADP/RL/optimal control - new
- Distributed computing in RL - new

AlphaZero/DRL: Anything new here?

"Single Deep Reinforcement Learning algorithm employs distributed computing to achieve superhuman performance in Go, Chess and Shogi."

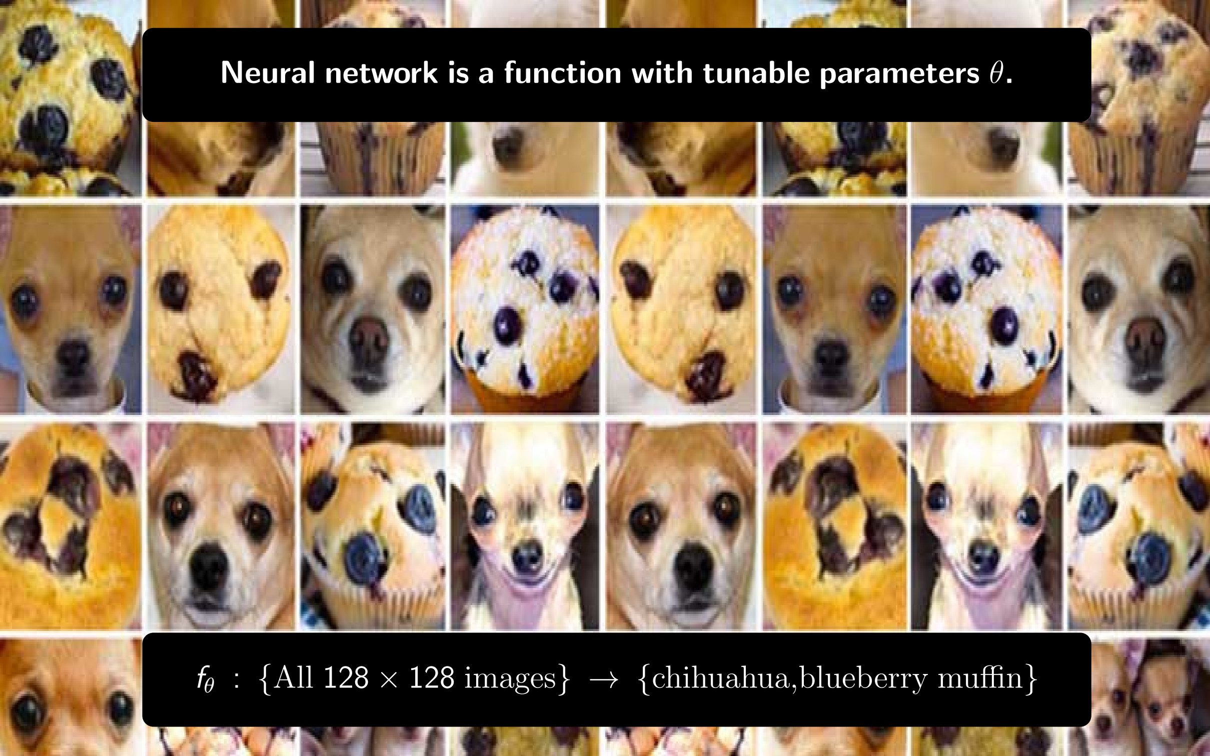
- Reinforcement Learning: Machine learning (regression) which fits a model on data obtained from a simulated environment - Existing field built upon earlier work in approximate dynamic programming & optimal control.
- Deep RL: using *deep* neural networks in ADP/RL/optimal control - new
- Distributed computing in RL - new
- Generic approach for problems considered difficult - exciting!

AlphaZero/DRL: Anything new here?

"Single Deep Reinforcement Learning algorithm employs distributed computing to achieve superhuman performance in Go, Chess and Shogi."

- Reinforcement Learning: Machine learning (regression) which fits a model on data obtained from a simulated environment - Existing field built upon earlier work in approximate dynamic programming & optimal control.
- Deep RL: using *deep* neural networks in ADP/RL/optimal control - new
- Distributed computing in RL - new
- Generic approach for problems considered difficult - exciting!

Neural network is a function with tunable parameters θ.

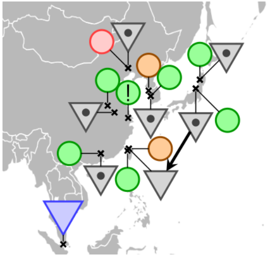


$$f_\theta \,:\, \{\text{All } 128 \times 128 \text{ images}\} \,\rightarrow\, \{\text{chihuahua,blueberry muffin}\}$$

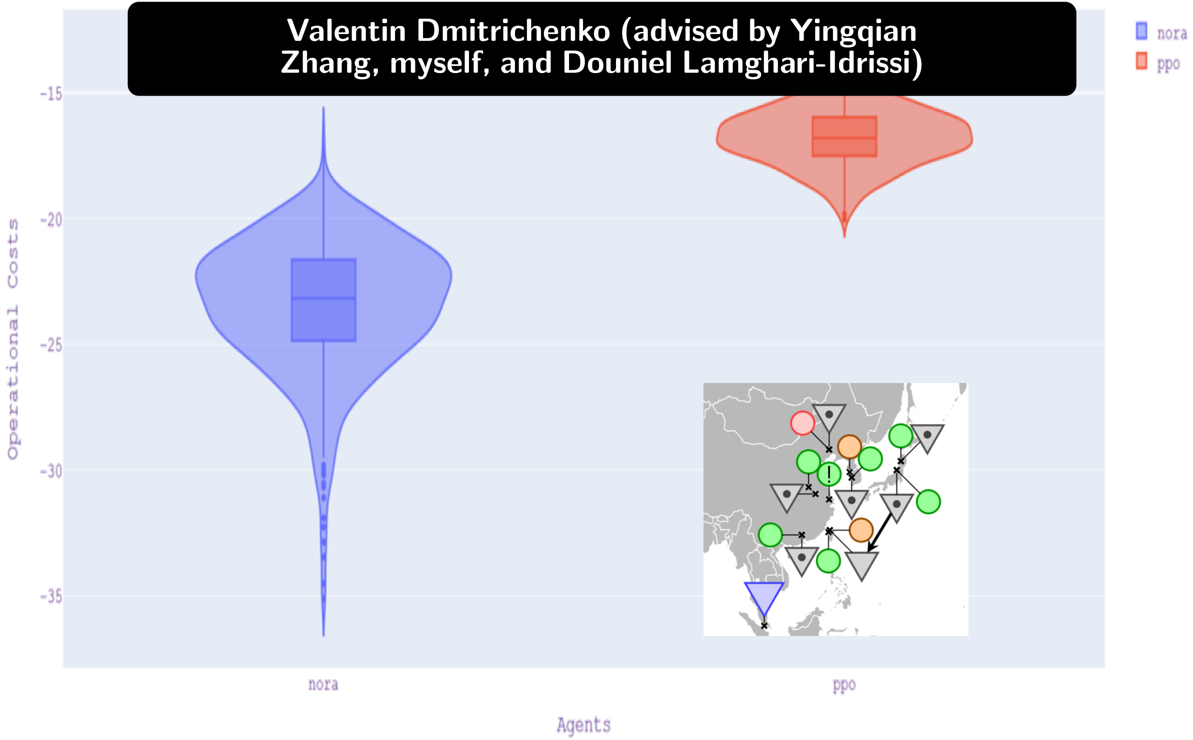
Results for Operations Management

- Results “before” DynaPlex.
- DynaPlex
- Results with Dynaplex.

Valentin Dmitrichenko (advised by Yingqian Zhang, myself, and Douniel Lamghari-Idrissi)

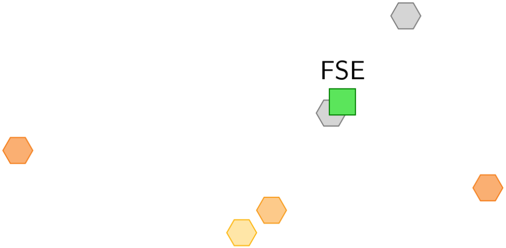


Valentin Dmitrichenko (advised by Yingqian Zhang, myself, and Douniel Lamghari-Idrissi)



Dynamic Traveling Maintainer Problem with Alerts

nQR-DDQN ($\mathbf{L}_0$)	M1-Q1	**16.365** [16.171, 16.56]	**124.96** [123.541, 126.378]	**32.804** [32.443, 33.165]
	M1-Q4	**8.806** [8.608, 9.004]	**47.408** [46.894, 47.922]	**14.513** [14.343, 14.683]
	M2-Q2Q3	**25.139** [24.935, 25.343]	**202.311** [200.675, 203.946]	**46.995** [46.609, 47.381]
	M4-Q2Q3	**92.654** [90.736, 94.571]	**470.625** [467.640, 473.611]	**106.525** [105.717, 107.334]
	M6-Q2Q3Q4	**176.642** [175.234, 178.051]	**711.188** [705.789, 716.586]	**159.527** [158.294, 160.760]
	M6-Q2Q3Q4-C		**347.500** [344.986, 350.014]	



DynaPlex

Observation: Publishing in OM requires beating very capable benchmarks!

DynaPlex

Observation: Publishing in OM requires beating very capable benchmarks!

Project with 11 industrial partners and 2 universities (TU/e, UT)

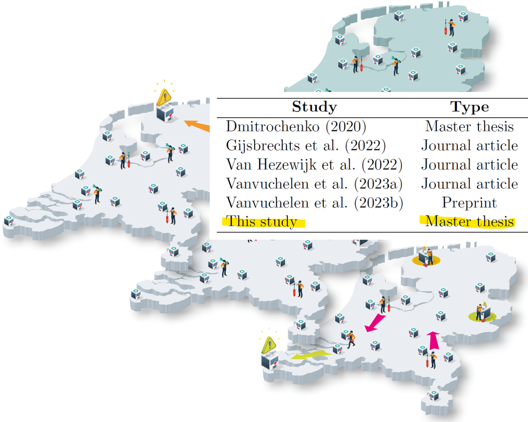
Martijn Mes, Fabian Akkerman, Remco Dijkman, Yingqian Zhang, Riccardo Lo Bianco, Tarkan Temizöz, Luca Begnardi, Peter Verleijsdonk, Simon Voorberg, etc.

Goal - DRL framework focused on applications in OM:

- Design choices focused on usability within OM.
- Range of algorithms : DCL, PPO, ADP-type, ...
- Suitable for super-computing.

DynaPlex results

DynaPlex results



Study	Type	Problem	#SKUs	#locations
Dmitrochenko (2020)	Master thesis	Fully connected network	1	5
Gijsbrechts et al. (2022)	Journal article	Distribution system	1	11
Van Hezewijk et al. (2022)	Journal article	Capacitated lot sizing	10	-
Vanvuchelen et al. (2023a)	Journal article	Fully connected network	1	10
Vanvuchelen et al. (2023b)	Preprint	Joint replenishment	40	1
This study	Master thesis	Fully connected network	10,000	60

Schedule

10:00	Welcome and introduction round (Willem)
10:15	The why: Algorithms for online decision making in OM (Willem)
11:00	Break
11:05	The how: Value-based reinforcement learning (Martijn)
12:15	Lunch
13:10	The how: Policy-based reinforcement Learning and Benchmarking (Willem)
14:00	Break
14:15	The how: Neural Network Architectures in DRL (Zaharah)
15:00	Discuss own problem and possible approaches (in groups with coaching)
15:45	Pitch problem and solution ideas (2-minute pitch per group)
16:15	End

Schedule

10:00 Welcome and introduction round (Willem)

10:15 The why: Algorithms for online decision making in OM (Willem)

11:00 Break

11:05 The how: Value-based reinforcement learning (Martijn)

12:15 Lunch

13:10 The how: Policy-based reinforcement Learning and Benchmarking (Willem)

14:00 Break

14:15 The how: Neural Network Architectures in DRL (Zaharah)

15:00 Discuss own problem and possible approaches (in groups with coaching)

15:45 Pitch problem and solution ideas (2-minute pitch per group)

16:15 End