Operations Research in the Audi Supply Chain

Using the example of the optimized delivery frequencies

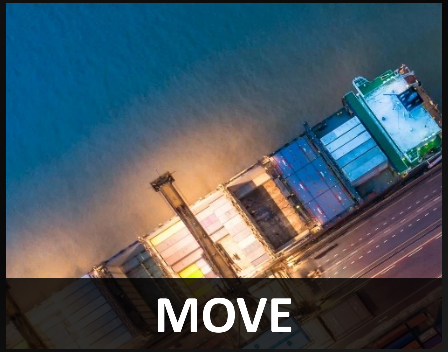
Sascha Tausch

The Supply Chain Team of Audi

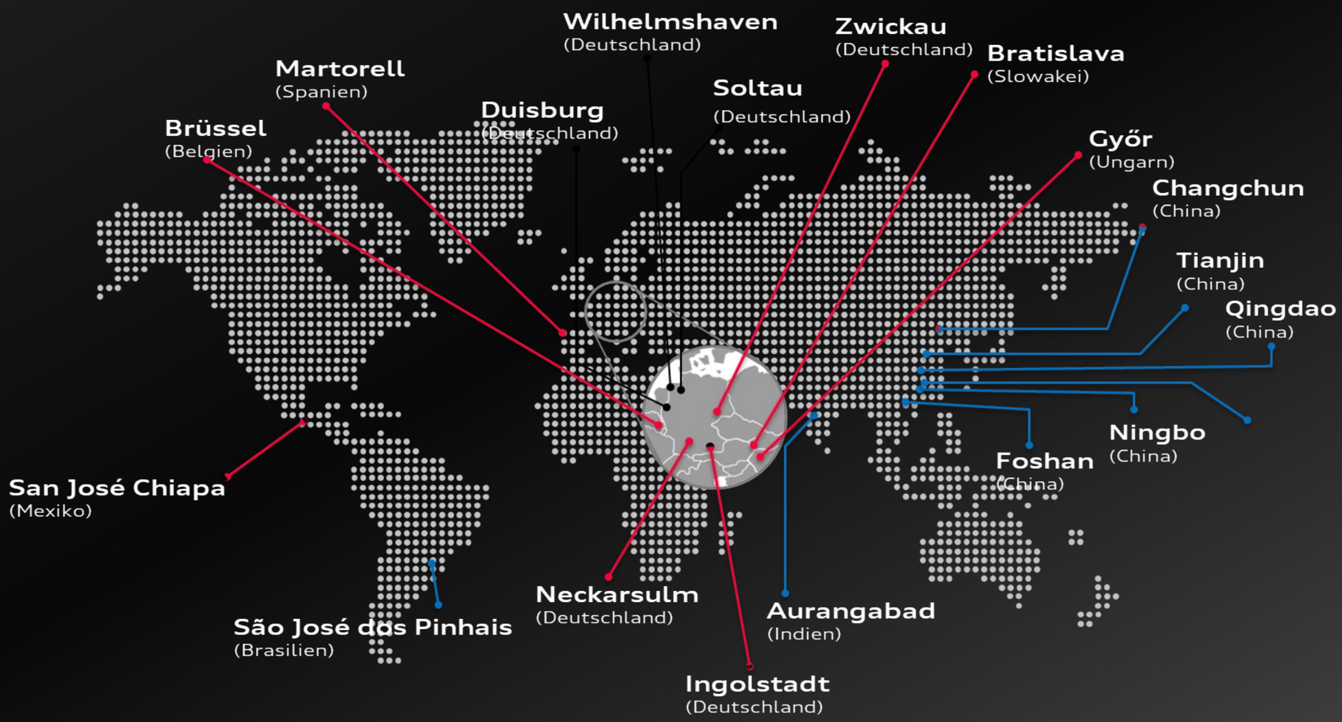
The Team Supply Chain

Supply Chain

We …



… at and for 17 sites worldwide.



FBU-Werk
xKD-Werk
Verpackungsstandort

Inbound
~1.500 trucks/day
~100 wagons/day

Commissioning
~ 1,2 mil. picks/day

Container-Transfers
~60 thous. GLT/day
~90 thous. KLT/day

xKD-Shipment
~ 4.500 m³/day

Logistics area
~1,5 mil. m²

Ext. Service Provider
~ 4.000

* Bezugsjahr 2022

Team smartDecisions

smartDecisions is a team of five mathematicians and engineers within the supply chain organization.

We translate complex planning problems into mathematical models and implement them in a full-stack fashion.

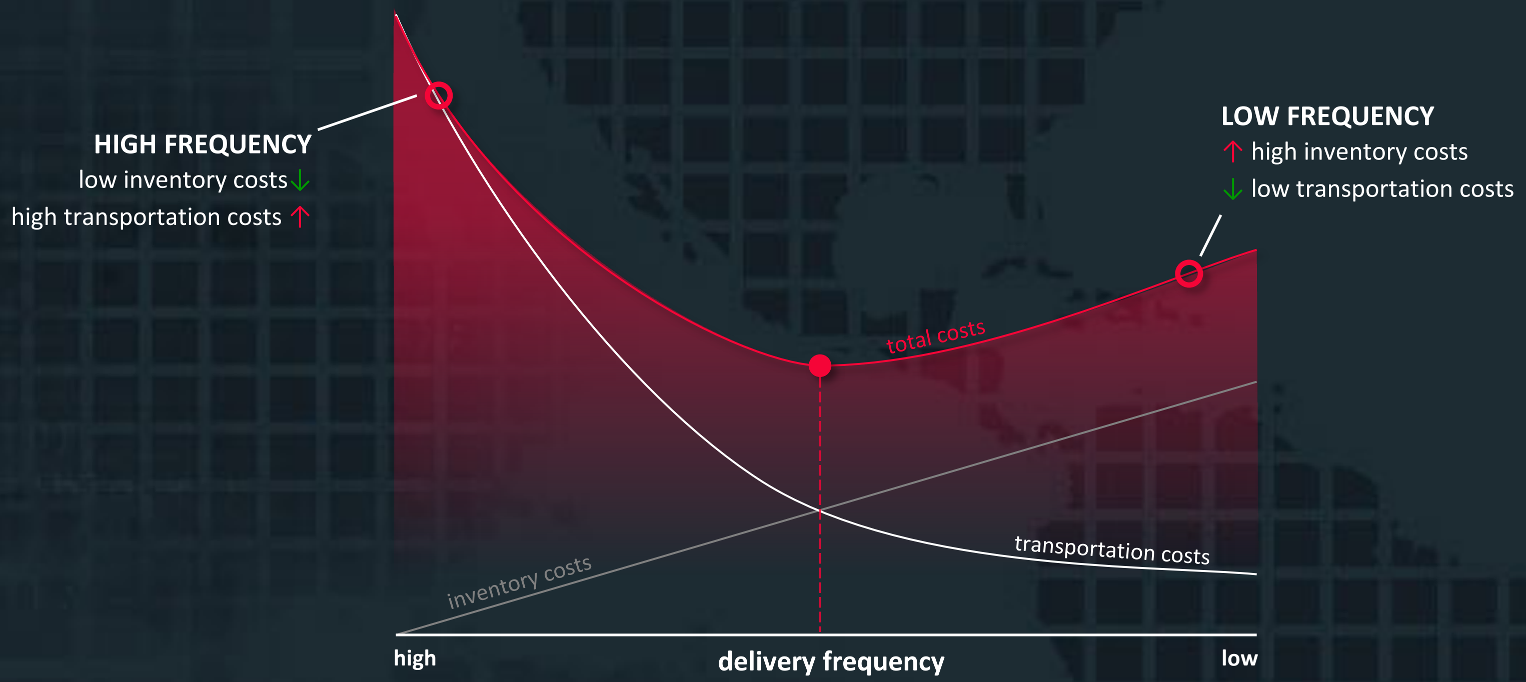
So far we developed over 30 productive models.

Example:
Delivery Frequency

DELIVERY FREQUENCY

The classical Economic Order Quantity (EOQ)[1][2]
defines the sweet spot
of transportation and inventory costs

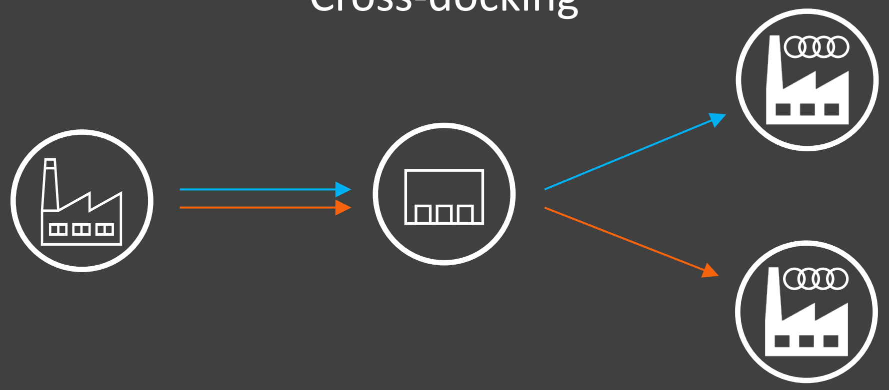
… but only for a single relation!



We distinguish between two concepts of transportation

Direct deliveries

Cross-docking

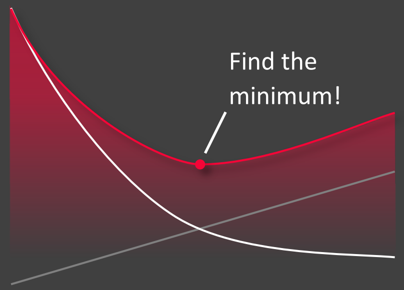


We distinguish between two concepts of transportation

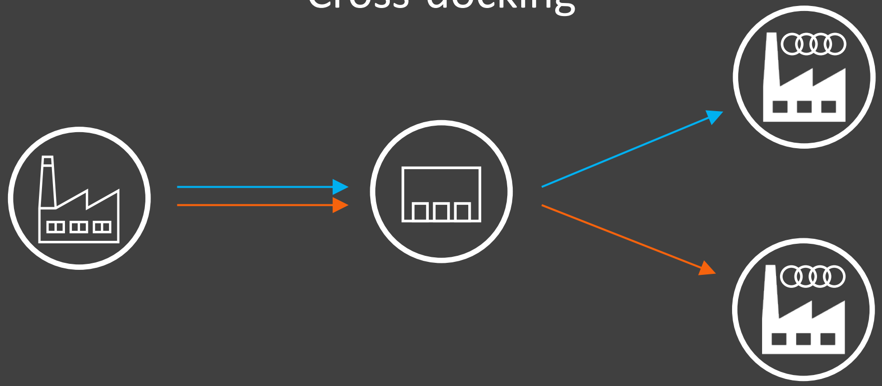
Direct deliveries

Directs can be optimized independently

… if there are no capacity restrictions



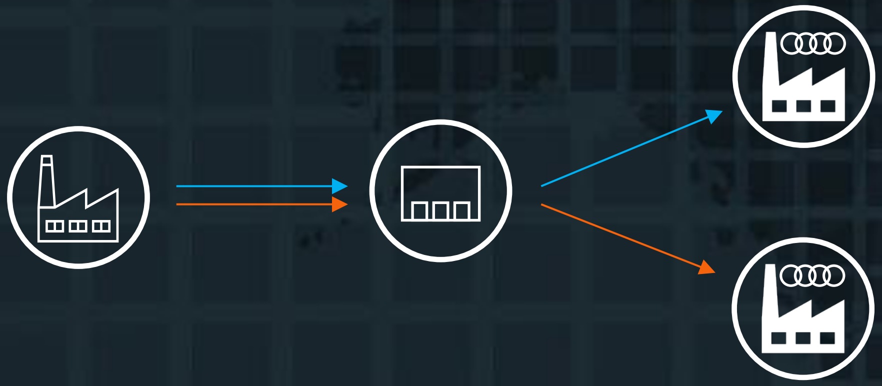
Cross-docking



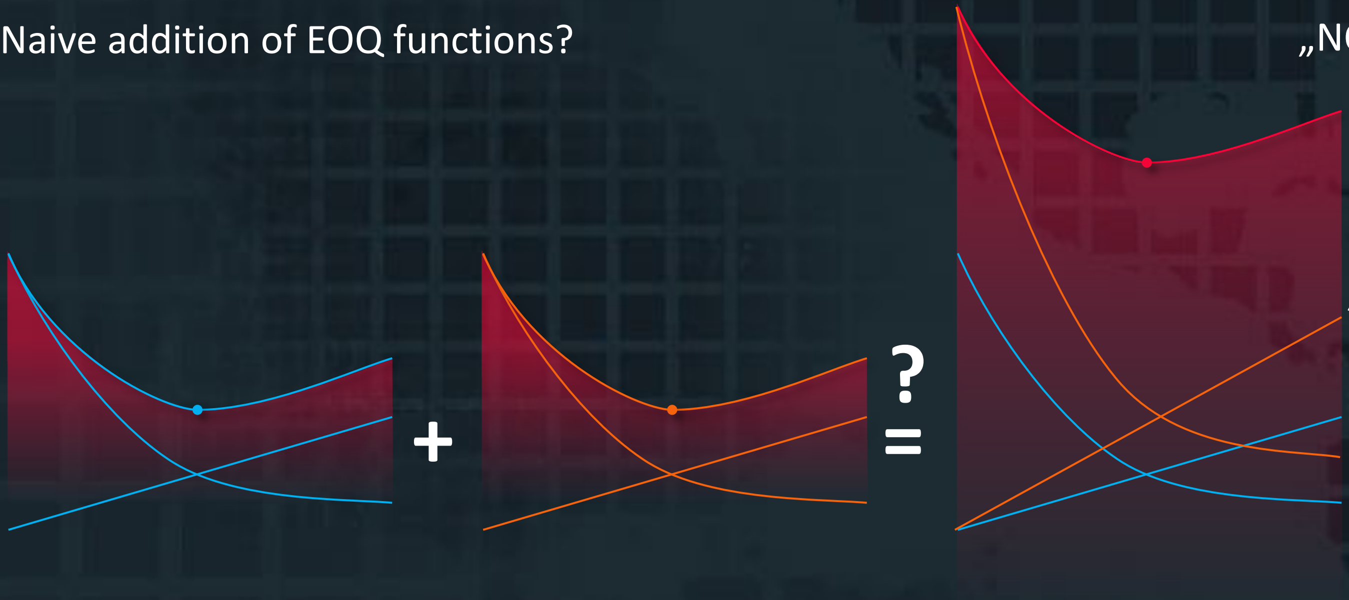
…whereas cross-docking is more complex

but with greater potential ☺

Cross-docking



Naive addition of EOQ functions?

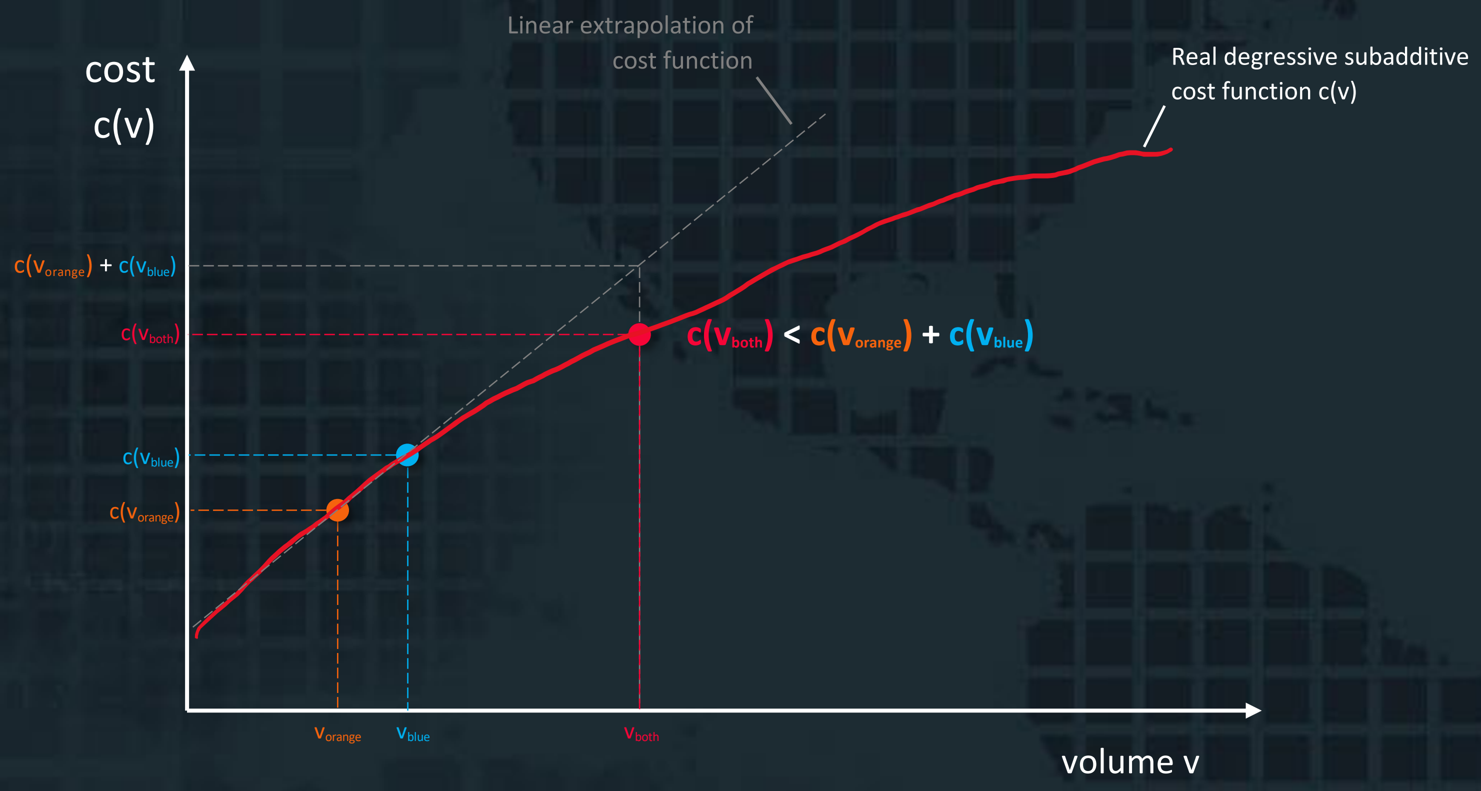


The inventory cost function one can simply add (it is not affected by cross-docking)

…wheras the transportation costs are subadditive: „The more you combine in a single freight, the cheaper the costs!“

Delivery schemes for each supplier-factory relation are predefined

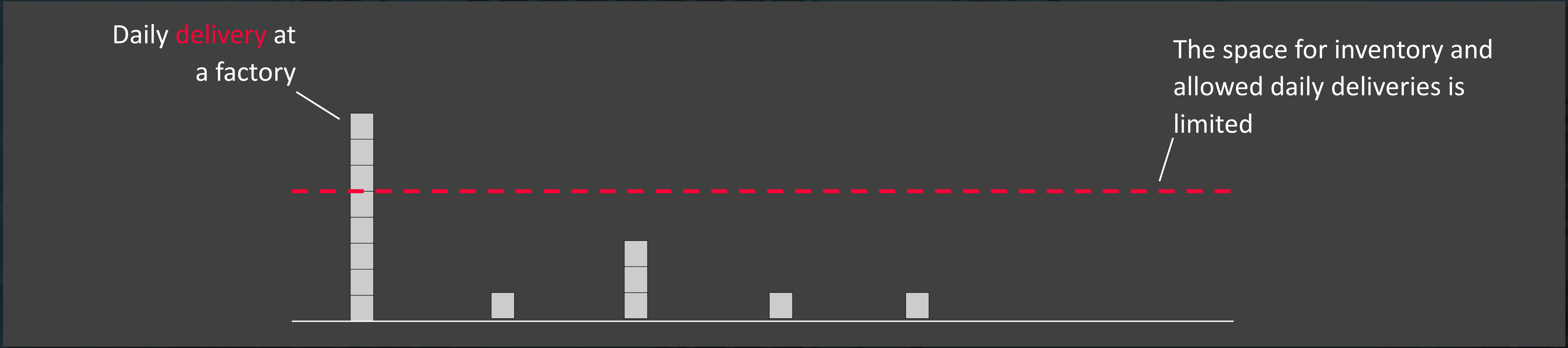
The real transportation costs are subadditive

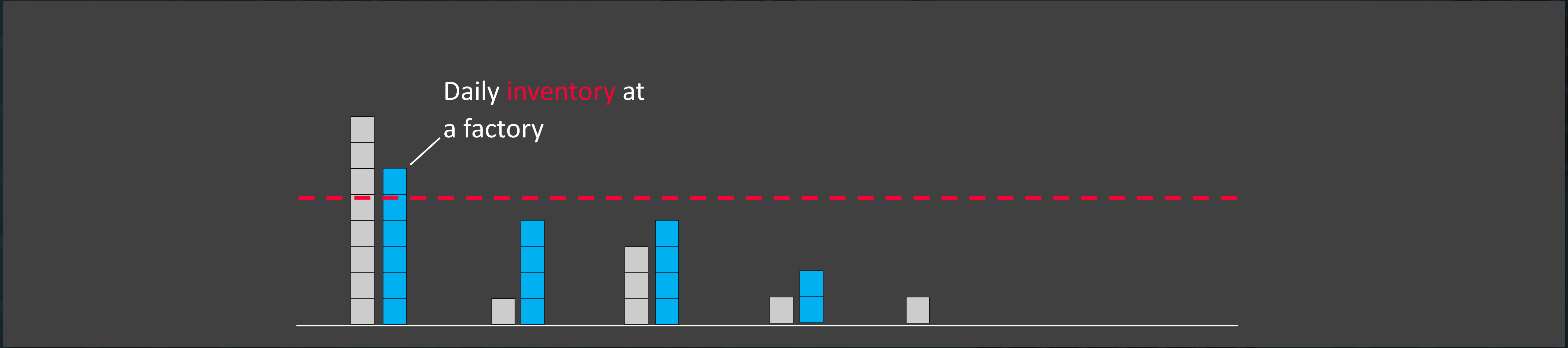


That was it?

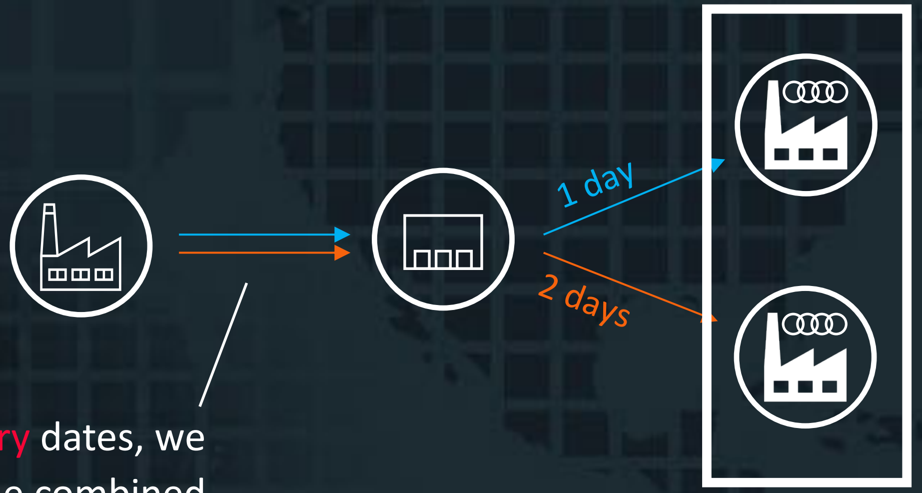
Unfortunately not…

We must also consider the factories delivery and inventory restrictions





We keep track between both delivery and arrival dates to aggregate from suppliers’ and factories’ perspectives



By knowing the delivery dates, we can minimize the combined transportation costs

By computing arrival dates and aggregating over all suppliers, we can calculate deliveries and inventories

And how can we model that?

State-of-the-art

[1] Harris (1913): ***How Many Parts to Make at Once***
Harris proposed the classic EOQ model. It defines batch sizes for productions with the lowest inventory and setup costs. He supposes inventory costs which grow linearly and setup costs which decline in a degressive way with higher batch sizes. The optimal batch size is where both cost functions intersect.

[2] Andler (1929): ***Rationalisierung der Fabrikation und optimale Losgrösse***
In the german-speaking countries the EOQ is often referred as `Andler-Formel´. In his dissertation he uses a slightly different way of computing the inventory. For further discussions see [6].

[3] Boysen et al (2014): ***Part logistics in the automotive industry: Decision problems, literature review and research agenda***
In his review paper Boysen et al. show important decisions problems and related models and publications for inbound and in-house part logistics. They cover the processes from the call order at a supplier over transport logistics, reception, storage, sequencing, delivery to and presentation at assembly lines, and the return of empty containers.

[4] Baller et al (2022): ***Optimizing automotive inbound logistics: A mixed-integer linear programming approach***
Baller et al. propose a mixed-integer-programming model which selects optimal transportation modes for a suppliers. It can chose between express services (CES), less than truckload (LTL), and full truck load (FTL). Furthermore the model defines exact delivery quantities for each supplier and each time period.

We focus on three aspects

1. Basic model formulation
2. Integration of CROSS DOCKING
3. Approximate the (unknown) TRANSPORTATION COSTS

Key requirements

Targets
Minimize the direct transportation costs
Minimize the shared transportation costs to cross-docks
Minimize the inventory costs

Decision Variables
Chosen delivery scheme per supplier and factory
Volume per supplier -> cross-dock/factory transport

Constraints
Activate one delivery scheme per supplier and factory
Limit the quantity of daily deliveries per factory
Limit the inventory per factory

Model draft

Indices and sets

Basics

T	List of all days	t ϵ T, T = [1, 2, …]
S	Set of suppliers	s ϵ S, S = {„Supplier A“, „Supplier B“, …}
F	Set of all factories	f ϵ F, F = {„factory A“, …}

Dependent Sets

F_s	Set of all delivered factories per supplier s
$D_{s,f}$	Set of all possible delivery schemes per supplier s and factory f

Dependend settings

$q_t(d)$	Quantity of daily deliveries per delivery scheme d
q_f^{max}	Maximimum allowed deliveries per factory f
$i_t(d)$	Quantity of daily inventory resulting from delivery scheme d
i_f^{max}	Maximimum allowed inventory per factory f
$v_t(d)$	Volume of daily deliveries per delivery scheme d
$c_s^{trans.}(v)$	Functions giving transport cost for a supplier and transported volume v
$c^{inv.}(d)$	Function giving inventory cost for a delivery scheme d
$\tau_{f,s}(d)$	Delay (in days) between supplier s and factory f

Model draft

Decision variables and target function

Decision Variables

chosen_delivery_scheme$_{s,f,d}$ $\in [0; 1]\ \forall s \in S,\ \forall f \in F_s,\ \forall d \in D_{s,f}$

volume$_{s,t}$ $\in \mathbb{R}_{\geq 0}\ \forall s \in S,\ \forall t \in T$

Target function

$$\min \sum_{s\in S, f\in F_s} c_s^{trans.}(\text{volume}_{s,t}) + \sum_{f\in F_s} \sum_{d\in D_{s,f}} c^{inv.}(d) * \text{chosen\_delivery\_scheme}_{s,f,d}$$

transportation costs **inventory costs**

Model draft
Inequalities

(1) Activate one delivery scheme per supplier and factory

$$\sum_{d \in D_{s,f}} \text{chosen\_delivery\_scheme}_{s,f,d} = 1 \qquad \forall s \in S, \forall f \in F_s$$

(2) Compute transported volume per delivery

$$\text{volume}_{s,t} = \sum_{f \in F_s} \sum_{d \in D_{s,f}} \text{chosen\_delivery\_scheme}_{s,f,d} * v_{t+\tau_{f,s}}(d) \qquad \forall t \in T, \forall s \in S$$

(3) Limit the quantity of daily deliveries per factory

$$\sum_{s \in S} \sum_{d \in D_{s,f}} \text{chosen\_delivery\_scheme}_{s,f,d} \cdot q_t(d) \leq q_f^{max} \qquad \forall t \in T, \forall f \in F$$

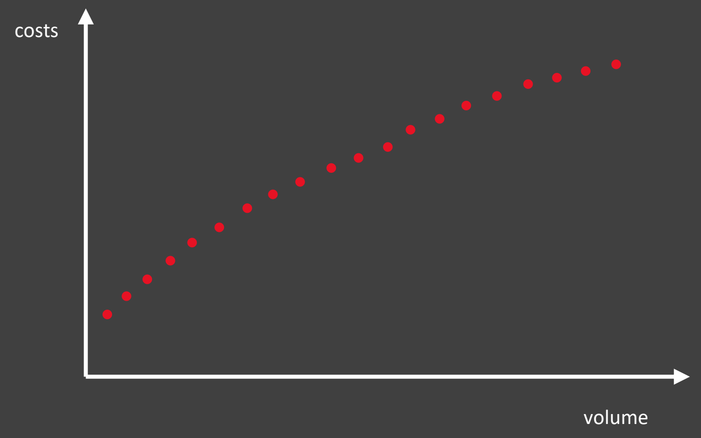
(4) Limit the inventory per factory

$$\sum_{s \in S} \sum_{d \in D_{s,f}} \text{chosen\_delivery\_scheme}_{s,f,d} \cdot i_t(d) \leq i_f^{max} \qquad \forall t \in T, \forall f \in F$$

How can we add the (unknown) subbaditive cost function c(v)?

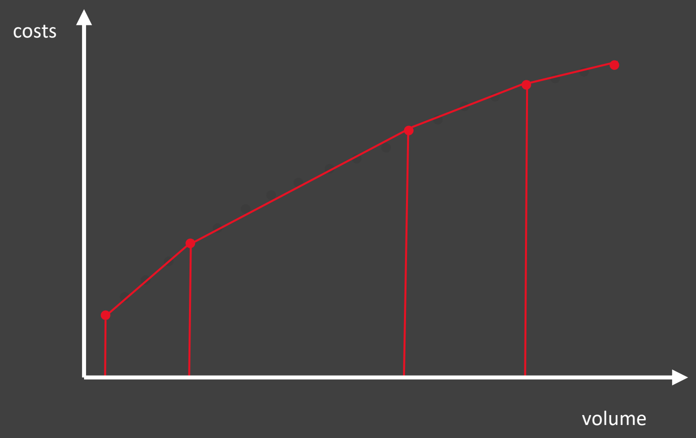
With a piecewise linear function!

Discretized costs



- We can query an oracle that gives us for any volume x the true costs f(x)
- We do that for a large number of points between supplier‘s min and max volume

Discretized approx. costs



- To improve model performance we reduce the number of points while keeping the resulting error low

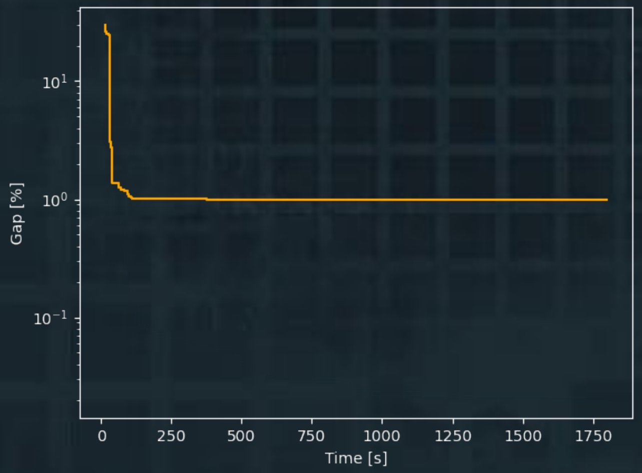
Mathematical challenges

Transportation Costs – Implementation Details

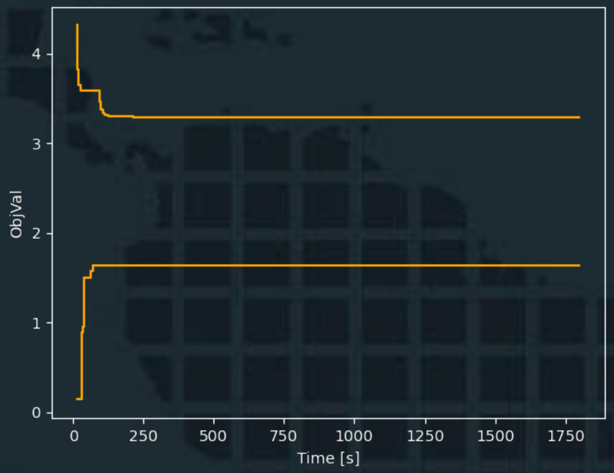
Gurobi allows piecewise-linear (pwl) modelling via its *general constraints* api

```
m = gp.Model()

x = m.addVars(suppliers, days)  # transport volume per supplier and day
y = m.addVars(suppliers, days)  # transport cost per supplier and day

for s in suppliers:
    for d in days:
        # set y = f(x)
        m.addGenConstrPWL(xvar=x[s, d], yvar=y[s, d], xpts=pwl_x, ypts=pwl_y)

m.setObjective(y.sum() + ...)
```

Although Gurobi includes a piecewise-linear simplex algorithm (cf. [5]), the long runtime led to suboptimal results, but…



100%



Mathematical challenges

Transportation Costs – Implementation Details

Gurobi allows piecewise-linear (pwl) modelling via its *general constraints* api

```
m = gp.Model()

x = m.addVars(suppliers, days)  # transport volume per supplier and day
y = m.addVars(suppliers, days)  # transport cost per supplier and day

for s in suppliers:
    for d in days:
        # set y = f(x)
        m.addGenConstrPWL(xvar=x[s, d], yvar=y[s, d], xpts=pwl_x, ypts=pwl_y)

m.setObjective(y.sum() + ...)
```

```
m = gp.Model()

x = m.addVars(suppliers, days)  # transport volume per supplier and day

m.setObjective(...)  # first set all the other objective terms

for s in suppliers:
    for d in days:
        # set cost directly in objective
        m.setPWLObj(x[s, d], pwl_x, pwl_y)
```

Thanks to Gurobi support team we improved the model's running time immensely

Model statistics

140.000 columns

150.000 rows

10.000.000 nonzeros

GUROBI 10.0.1

8core CPU Intel(R) Xeon(R) @ 2.70GHz

→ 5% Gap after total 10min

Final Results

INVENTORY:

0 -1% ↑

TRANSPORTATION:

20-30% ↓

References

[1] Ford W. Harris (1913): *How Many Parts to Make at Once*. The Magazine of Management 10(2): 135-136,152

[2] Kurt Andler (1929): *Rationalisierung der Fabrikation und optimale Losgrösse*. München: R. Oldenbourg

[3] Boyen et al. (2014): *Part logistics in the automotive industry: Decision problems, literature review and research agenda.* European Journal of Operational Research 242(1), 107-120

[4] Baller et al. (2022): *Optimizing automotive inbound logistics: A mixed-integer linear programming approach*. Transportation Research Part E: Logistics and Transportation Review 163

[5] Robert Fourer (1985): *A simplex algorithm for piecewise-linear programming I: Derivation and proof.* Mathematical Programming 33, 204–233

[6] Georg Krieg (2005): *Neue Erkenntnisse zu Andlers Losgrößenformel.* Arbeitspapier, Katholische Universität Eichstätt-Ingolstadt

Thank you for your attention!