Bootstrap-Enhanced Distributionally Robust Scheduling of Loading Yards

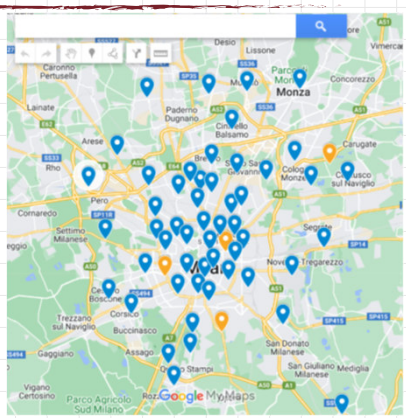
Vittorio Maniezzo, University of Bologna
Livio Fenga, University of Exeter

Agenda

1) The theory behind BESO (Bootstrap Enhanced Scenario Optimization)
2) The triggering use case: city logistics
3) The scheduling use case: loading yards

First problem

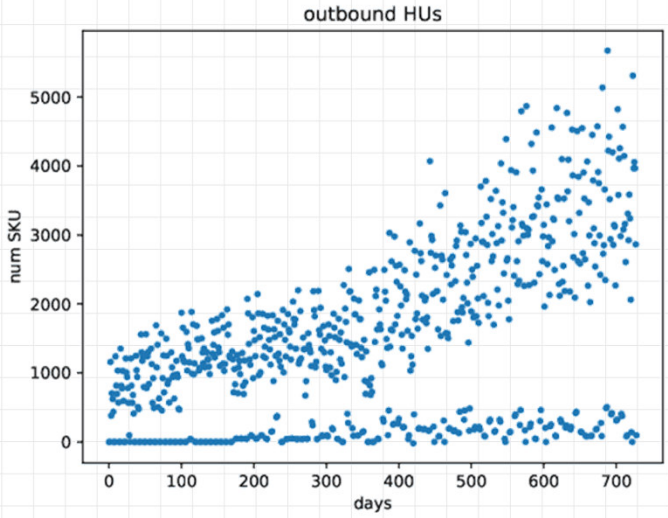




- 52 retailer stores
- 4 DCs: one big, one smaller, 2 satellites close to city center

Allocation for Christmas season! Preliminary analysis during summer.

Second problem



The theory

Prequel: Stochastic progr. and SAA

Decisions often precede the observation of uncertain quantities.

Stochastic programming describes uncertainty through a probability distribution P.

Two-stage setting: the first-stage decision x precedes the realization of uncertain parameters ξ, the recourse decision y adapts to it.

A common objective is to minimize expected total cost: first-stage cost plus expected recourse cost $min_x c(x) + \mathbb{E}_P[Q(x, \xi)]$.

Sample Average Approximation (SAA):
P → n sampled scenarios, probability $1/n$
empirical distribution P_n,
expectations → sample averages,

Standard SAA can disappoint out of sample, especially if n is small.

Prequel: Robust optimization

Uncertain parameters ξ belong to an uncertainty set $\mathcal{U}$, specifying the realizations against which protection is required.

No probability distribution over these realizations is specified.

Decisions minimize worst-case cost while remaining feasible for every $\xi \in \mathcal{U}$.

$$\min_{x \in X} \sup_{\xi \in \mathcal{U}} f(x, \xi)$$

$$\text{subject to } g_i(x, \xi) \leq 0 \; \forall \xi \in \mathcal{U}.$$

→ worst-case objective and feasibility across the uncertainty set.

Enlarging $\mathcal{U}$ strengthens protection but increases the optimal cost: the uncertainty set controls the degree of conservatism

Prequel: Distributionally robust optimization and ambiguity sets

DRO: uncertainty about the probability distribution itself, not just the random parameters.
An ambiguity set $\mathcal{P}$ is the set of plausible distributions.

DRO minimizes the worst-case expected cost over $\mathcal{P}$.

$$min_x sup_{P \in \mathcal{P}} \mathbb{E}_P[Q(x, \xi)].$$

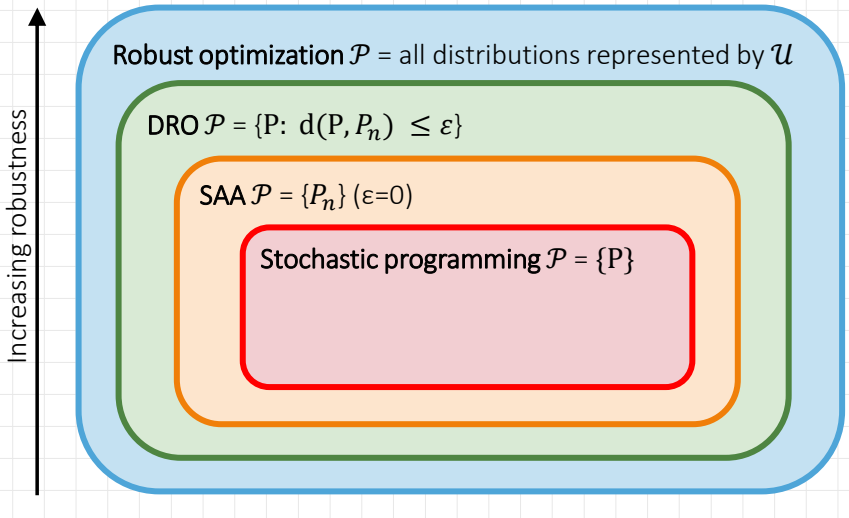
Two common ways to construct $\mathcal{P}$:

- moment-based: support restrictions and moment constraints;
- discrepancy-based: all P within radius ε of P_n, measured by a Wasserstein distance or a ϕ-divergence.

DRO bridges the two prequels:

- $\mathcal{P}$ = {P} → *stochastic programming* (no distributional uncertainty),
- $\mathcal{P}$ = $\{P_n\}$ (ε=0) → *SAA* (empirical distribution is taken as the truth);
- DRO
- $\mathcal{P}$ = all distributions already supported on $\mathcal{U}$ → *robust optimization* (worst case over the support).

Distributional ignorance



Smart Predict-then-Optimize (SPO)

SPO uses predictions as inputs to an optimization model.

Conventional prediction models minimize statistical forecast error
→ small forecast errors do not necessarily imply good decisions.

SPO uses a tractable surrogate loss to train predictions according to their decision consequences (*Elmachtoub, Grigas, 2017*).

Decision awareness enters through the predictor's training objective: it does not optimize over an ambiguity set of distributions.

BESO, generalities

The BESO framework integrates forecasting and optimization through bidirectional coupling:

- the **forecasting** component embeds optimization-relevant structure (without modifying the forecasting loss function),
- the **optimization** component is built on forecast outputs and their uncertainty characterization.

Appropriate when innovation noise and random shocks do not dominate forecast errors:
→ the forecasts used by the optimizer may be inaccurate but are usable if their errors are not systematically biased.

The two components can be integrated at different levels, resulting in two models: BESO-SAA and BESO-DRO

Univariate time series forecasting

Three widely used families of forecasting methods are:

Statistical models specify explicit structures for temporal dependence and uncertainty.

- Examples include ARIMA/SARIMA, ETS, and structural state-space models.
- Their assumptions about dynamics and errors depend on the model.

Neural models learn flexible, potentially nonlinear relationships within a chosen architecture.

- Examples include MLPs, LSTMs, GRUs, temporal CNNs, and Transformers.

Tree-based ensembles combine decision trees using lagged observations and derived features.

- Examples include Random Forests and gradient boosting methods such as XGBoost and LightGBM.

No approach consistently outperforms all others. We can use any of them.

Resampling, bootstrapping

Data-driven distribution estimation is useful when stationarity or known distributional forms cannot be assumed.

It is well suited to complex, noisy, or data-limited real-world settings.

Bootstrap (*Efron, 1979*): **generates empirical distributions** by repeatedly sampling from the data with replacement.

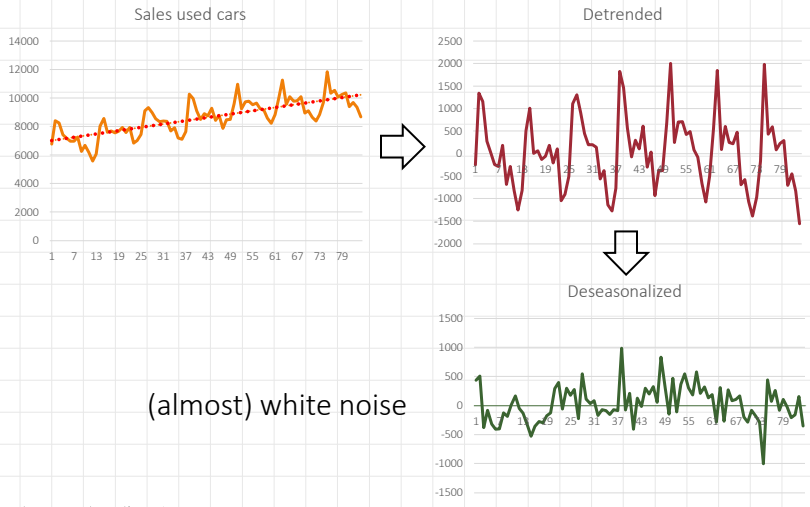
MEB (*Vinod, 2004*): **produces bootstrap replicates** that preserve time-series dependence and maximizes an information entropy function defined over the resampled data.

We use a predictive model fitted to each MEB replicate, one forecast trajectory per replicate.

The set of trajectories forms the scenario pool.

Prewhitening

Economics-related time series can often be effectively decomposed using STL



Recoloring

The prewhitened series only retains residuals unexplained by STL (i.e. *innovation noise and random shock*).

The observed residuals are realizations of a random variable whose distribution can be identified. (*hopefully normally distributed*).

Using the distribution of the residual series, we can generate alternative samples that are compatible with the overall model, i.e. samples that could have occurred.

Bootstrapping, how we did it

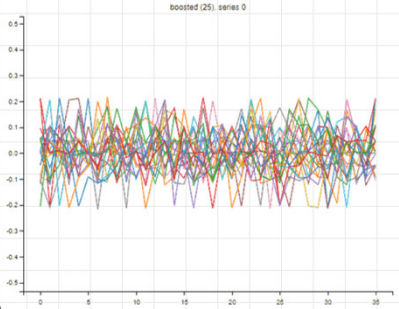
We fit an AR(p) model on each series

We extracted residuals ε_t, which were *sufficiently (?)* stationary

We resampled the residuals in ε_t^* and simulated new series as:

$$x_t^* = \phi_1 x_{t-1}^* + \ \ldots + \phi_p x_{t-p}^* + \varepsilon_t^*$$

We bootstrapped the residual series and reconstructed alternative series.



Entropy tilting

The bootstrapped trajectories define probability distributions.

They are assigned probability weights whose values are **influenced by the downstream optimization function**

Given a set of MEB-generated trajectories (scenarios) and associated features f_s and g_s, BESO defines scenario weights,

$$p_s^{OA} = \frac{\exp(\mu f_s + \gamma g_s)}{\sum_j \exp(\mu f_j + \gamma g_j)}$$

The features f_s and g_s are problem-dependent and are computed **before** the final optimization,

- f_s are computed directly from instance data,
- g_s approximate the contribution to the final cost of scenario s.

The measure $\hat{P}_n = \sum_s p_s^{OA} \delta_{\tau^s}$ is a cost-aware reference distribution

The prescriptive component

The set of bootstrap trajectories admits two distinct interpretations.

- a scenario-based formulation corresponds to SAA optimization under the cost-aware reference distribution $\hat{P}_n^{OA}$
- a worst-case formulation over the ambiguity set induced by the predictive distributions provides the DRO extension

Notationally,

- $x \in X$ denote the first-stage decision
- $\xi \in \Xi$ represent the uncertain problem parameters.
- the total cost is $c(x, \xi) = h(x) + R(x, \xi)$ with $h(x)$ the first-stage cost and $R(x, \xi)$ the recourse cost.
- $\tau^1, \ldots, \tau^n \in \Xi$ denote the generated model-based scenarios.
- Expected costs are $\mathbb{E}_P[c(x, \xi)]$,
- Scenario costs are $c(x, \tau^s)$.

BESO-SAA

In the first interpretation, the trajectories are regarded as alternative realizations of a single predictive distribution.

Given n trajectories $\tau^s, s \in S$, assigning objective-aware probability to each of them yields the empirical distribution

$$\hat{P}_n^{OA} = \sum_s p_s^{OA} \delta_{\tau^s}$$

The expected downstream cost is approximated by its weighted average over the bootstrap trajectories, leading to a **weighted sample average approximation** (SAA) variant model.

The first-stage decision x is common to all scenarios, while recourse decisions are scenario-specific.

The bootstrap procedure represents uncertainty within one estimated predictive distribution.

BESO-DRO

A second interpretation takes the expectation within each bootstrap distribution, and the worst case across bootstrap distributions.

We repeatedly resample d trajectories from S according to p_s^{OA}, each set of d is interpreted as a set of realizations of a different distribution.

If n_{ks} is the number of times series s appears in resample k, upon defining $w_{ks} = n_{ks}/d$ as the relative frequency of the series in resample k, we can define a collection of empirical distributions

$$\hat{P}^k = \sum_s w_{ks} \delta_{\tau^s} \qquad k \in K$$

where $K = \{1, \dots, m\}$, m being the number of resampled empirical distributions.

These distributions are then treated as alternative plausible descriptions of the underlying uncertain process and used to construct a related ambiguity set as

$$\mathcal{U}_{md} = conv\{\hat{P}^1, \dots, \hat{P}^m\}$$

The resulting formulation minimizes cost under the worst-case distribution in $\mathcal{U}_{md}$,

This model is distributionally robust with respect to reweightings of the scenario pool.

BESO-DRO

Let $C_k(x) = \mathbb{E}[c(x,\xi)] = \sum_s w_{ks} c(x,\tau^s)$ denote the expected optimized cost associated with bootstrap distribution $\hat{P}^k$.

Because expectation is linear in P, the supremum over the convex hull occurs at an extreme point, $\hat{P}^h$.

Therefore $sup_P \mathbb{E}_P[c(x,\xi)] = max_k \sum_s w_{ks} c(x,\tau^s)$

The DRO objective becomes

$$min_x max_k \sum_s w_{ks} c(x,\tau^s)$$

Where the w_{ks} are computed before the final optimization and passed to a solver as fixed numerical coefficients.

Increasing d: reduce distributional spread, BESO-DRO → BESO-SAA

Increasing m: enlarges ambiguity set, BESO-DRO → robust optimization

BESO-DRO

The construction procedure for BESO-DRO is:

1. Retain n trajectories τ^s, s ∈ S.
2. Compute their cost-aware base probabilities p_s^{OA}.
3. For each resampled distribution $k \in K$, draw d trajectories with replacement according to p_s^{OA}.
4. Treat that resample as an empirical distribution $\hat{P}^k$.
5. Define the ambiguity set as the convex hull of these distributions.

BESO decision models

1. Deterministic (baseline):

$$min_x c(x, \hat{\xi})$$ with $\hat{\xi}$ the bagged bootstrap forecast

2. BESO-SAA, weighted SAA variant:

$$min_x \sum_s p_s^{OA} c(x, \tau^s)$$

3. BESO-DRO:

$$min_x max_k \sum_s w_{ks} c(x, \tau^s)$$

use case 1:

City Logistics

The first case study

Large-scale organized distribution is often structured according to a two-echelon model, in which distribution centers (DCs) are used to store and forward goods to replenish the inventories of the final retailer's stores.

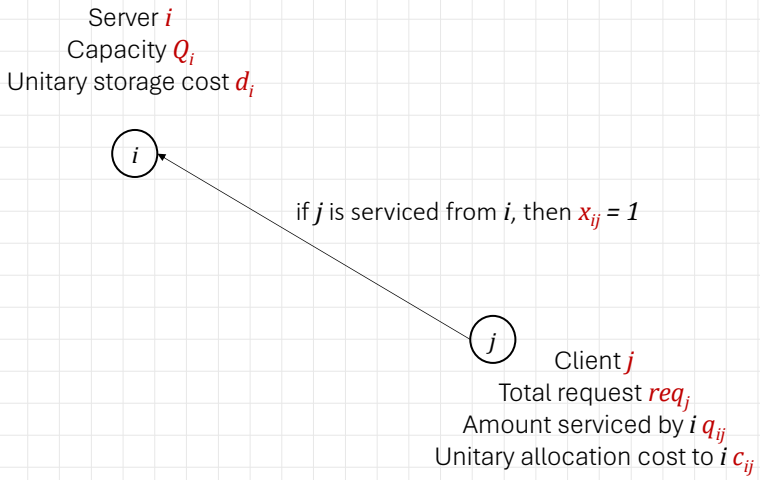
Stores are typically uniquely allocated to DCs for products of the same line, and predictive knowledge of future demand is essential to optimize the distribution process.

In the real-world case study:

- city retailer store demands were predicted based on individual sales time series and
- used to optimize the allocation of stores to DCs in a tactical planning context.

Our problem required ***dimensioning*** multiple DCs and assigning stores to them to meet expected demand during a critical period.

Toward a formulation



Deterministic allocation (SAP)

$$z(D) = min \quad \sum_{i\in I}\sum_{j\in J}(c_{ij}x_{ij} + d_i q_{ij}) \qquad (1)$$

subject to

$$\sum_{i\in I} q_{ij} = req_j \qquad j \in J \qquad (2)$$

$$\sum_{j\in J} q_{ij} \le Q_i \qquad i \in I \qquad (3)$$

$$\sum_{i\in I} x_{ij} \le b_j \qquad j \in J \qquad (4)$$

$$q_{ij} \le req_j\, x_{ij} \qquad i \in I, j \in J \qquad (5)$$

$$x_{ij} \in \{0,1\} \qquad i \in I, j \in J \qquad (6)$$

$$q_{ij} \in \mathbb{Z}_0^+ \qquad i \in I, j \in J \qquad (7)$$

Stochastic allocation

$$z(H) = min \quad \sum_{i\in I}\sum_{j\in J}(c_{ij}x_{ij} + d_i q_{ij}) \qquad (1)$$

subject to

$$\sum_{i\in I} q_{ij} = \hat{\rho}_j \qquad j \in J \qquad (2)$$

$$\sum_{j\in J} q_{ij} \le Q_i \qquad i \in I \qquad (3)$$

$$\sum_{i\in I} x_{ij} \le b_j \qquad j \in J \qquad (4)$$

$$q_{ij} \le \hat{\rho}_j\, x_{ij} \qquad i \in I, j \in J \qquad (5)$$

$$x_{ij} \in \{0,1\} \qquad i \in I, j \in J \qquad (6)$$

$$q_{ij} \in \mathbb{Z}_0^+ \qquad i \in I, j \in J \qquad (7)$$

Entropy tilting

For this use case, we considered no instance-specific tilting feature and defined the g_s as costs of a simplified transportation problem accounting for scenario demands and DC capacity constraints.

$$z(s) = \textit{min} \quad \sum_{ij} c_{ij} q_{ij}^s \qquad (1)$$

$$\textit{subject to} \quad \sum_{i \in I} q_{ij}^s = \rho_j^s \qquad j \in J, s \in S \qquad (2)$$

$$\sum_{j \in J} q_{ij}^s \leq Q_i \qquad i \in I, s \in S \qquad (3)$$

$$q_{ij}^s \in \mathbb{Z}_0^+ \qquad i \in I, j \in J, s \in S \qquad (7)$$

BESO-SAA

Fit an AR(p) model on each MEB series, independently.

Each model is used to generate a scenario $s \in S$, specifically, a realization ρ_j^s of the forecasted requests of customer $j \in J$ according to scenario S.

All scenarios were weighted by the corresponding p_s^{OA}.

The resulting SAA formulation is passed on to a MIP solver.

The solution details the server *i* that each client *j* will be associated with, as well as the amount q_{ij} it will request from the server.

It is therefore possible to estimate the actual capacity requested from each server.

BESO-SAA

$$z(SAA) = \textit{min} \sum_{ij} c_{ij} x_{ij} + \sum_{s \in S} p_s^{OA} \left(\sum_{ij} d_i q_{ij}^s + M \sum_j \epsilon_{js} \right) \quad (1)$$

subject to

$$\sum_{i \in I} q_{ij}^s + \epsilon_{js} = \rho_j^s \qquad j \in J, s \in S \quad (2)$$

$$\sum_{j \in J} q_{ij}^s \leq Q_i \qquad i \in I, s \in S \quad (3)$$

$$\sum_{i \in I} x_{ij} = b_j \qquad j \in J \quad (4)$$

$$q_{ij}^s \leq \rho_j^s \, x_{ij} \qquad i \in I, j \in J, s \in S \quad (5)$$

$$x_{ij} \in \{0,1\} \qquad i \in I, j \in J \quad (6)$$

$$q_{ij}^s \in \mathbb{Z}_0^+ \qquad i \in I, j \in J, s \in S \quad (7)$$

$$\epsilon_{js} \in \mathbb{Z}_0^+ \qquad j \in J, s \in S \quad (8)$$

BESO-DRO

The DRO model requires resampling among scenarios and moving the objective function into constraints so to implement the min / max objective:

$$z(DRO) = \quad min \sum_{ij} c_{ij} + z \quad (9)$$

$$\textit{subject to} \quad z \geq \sum_{s \in S} w_{ks} \left(\sum_{ij} d_i q_{ij}^s + M \sum_j \epsilon_{js} \right) \quad k \in K \quad (10)$$

(2) ... (8)

Tests, forecasts

MEB bootstrap sets were aggregated (bagged) to obtain forecasts.

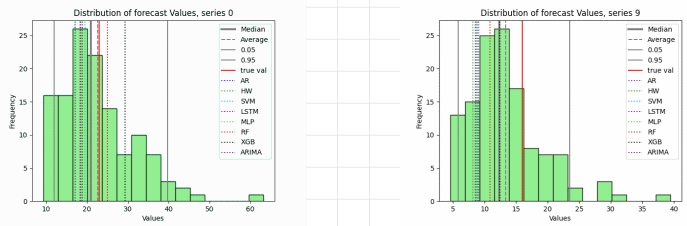
Besides MEB bagging we also considered alternative forecasts modules based on:

- AR
- Holt-Winters
- SARIMA
- Multilayer perceptrons
- LSTM networks
- Random forests
- XGboost

Results, forecasts

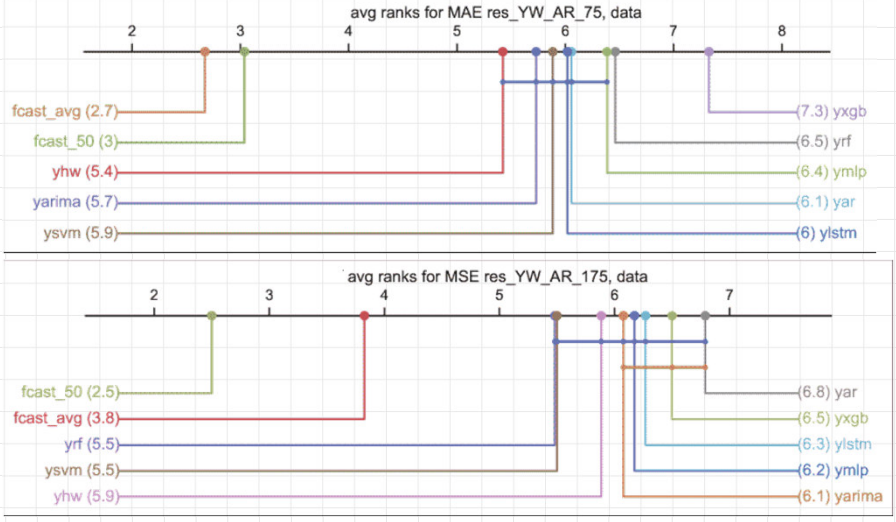
MEB based forecasting. Average rankings for different boostset sizes.

	YW_AR_75		YW_AR_125		YW_AR_175	
	MSE	**bias**	**MSE**	**bias**	**MSE**	**bias**
fcast_avg	2.67	0.08	2.58	0.05	3.83	0.04
fcast_50	3.04	0.06	3.19	0.04	2.50	0.03
yhw	5.42	0.08	6.06	0.04	5.88	0.03
yarima	5.73	0.05	6.04	0.03	6.08	0.02
ysvm	5.88	0.08	5.50	0.05	5.50	0.03
ylstm	6.02	0.06	5.52	0.04	6.27	0.02
yar	6.06	0.04	6.54	0.02	6.79	0.02
ymlp	6.38	0.05	5.98	0.03	6.17	0.02
yrf	6.46	0.11	6.31	0.06	5.48	0.04
yxgb	7.33	0.12	7.29	0.07	6.50	0.05



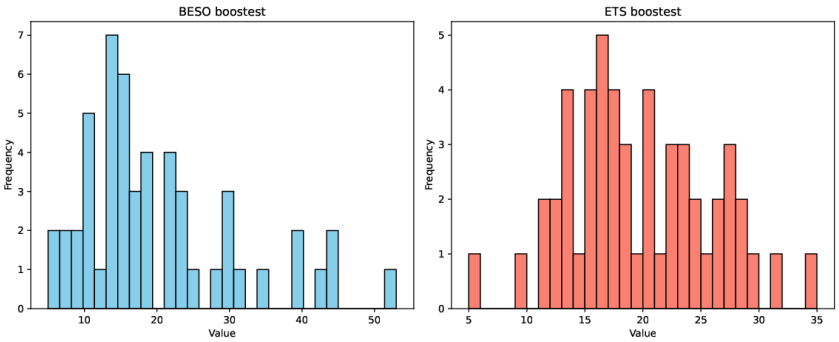
Results, forecasts

Critical difference diagrams (*Fawaz et al., 2019*)



Results, comparison

BESO vs. distribution-based, gaussian derived from confidence intervals of Holt-Winters ETS forecasts.



MEB: wider range and higher skewness

Results, optimization

	ETS	SAP	BESO-SAA	ETS-SAA	BESO-DRO	WEK
Assign. Cost	19682	20604	20529.77	19918.27	20535.44	20516.42
Optimality gap%	0	0	0	0	0	0
Infeasibilities	0	0	0	0	0	0
num.variables	416	416	19708	19708	19709	19784
num.constraints	316	316	19852	19852	19952	25477
num.non zeros	1040	1040	66508	66508	565184	662683
t.Cpu	0.17	0.05	10.72	4.14	21.79	55.26

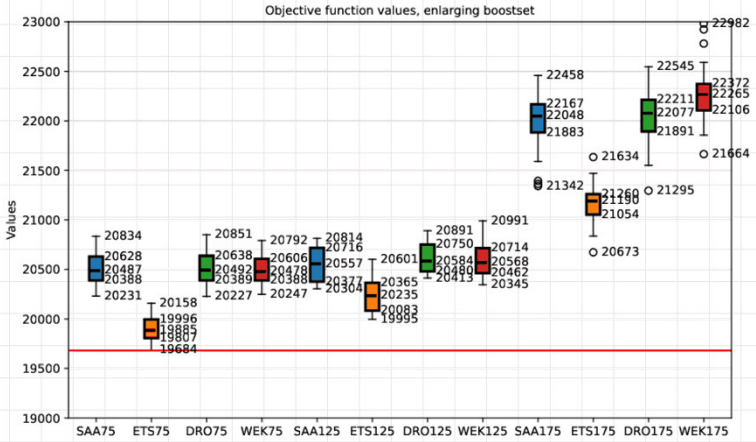
Table 2: Solution and model data on the use case instance and 75 boostrap set

	ETS			SAP			ETS-SAA			BESO-SAA			WEK			BESO-DRO		
	exp	true	ovr	exp	true	ovr	peak	true	ovr	peak	true	ovr	peak	true	ovr	peak	true	ovr
DC 0	119	115	0	120	111	0	120	118	0	120	111	0	120	111	0	120	111	0
DC 1	339	431	0	482	459	0	481	436	0	483	462	0	481	459	0	481	459	0
DC 2	277	304	4	300	288	0	298	299	0	300	289	0	298	288	0	298	288	0
DC 3	166	173	0	180	180	0	165	170	0	180	161	0	180	165	0	180	165	0

Table 3: DC usage (expected, realized demand, realized overload per method)

BESO: same solution than Wasserstein-based, formulation fairly simpler

Results, optimization



The cost is essentially stable in the cases of the 75 and 125 series sets but increases significantly in the case of the 175 boostset.

This is due to some extreme scenarios appearing in the forecasts.

Comments, 1st use case

BESO leads from forecasts to decisions that account for forecast uncertainty

- Bootstrapped forecasts generate realistic future scenarios directly from data, even from short, irregular histories
- Scenarios are weighted by their cost impact and fed into stochastic (SAA) or robust (DRO) optimization
- Every step is transparent and explainable: forecast, scenarios, weights, decision
- Proof of concept: DC-to-warehouse allocation and capacity sizing, where BESO proved competitive against the state of the art.

Takeaway: wherever decisions rest on uncertain forecasts, BESO turns that uncertainty into reliable decisions.

use case 2:

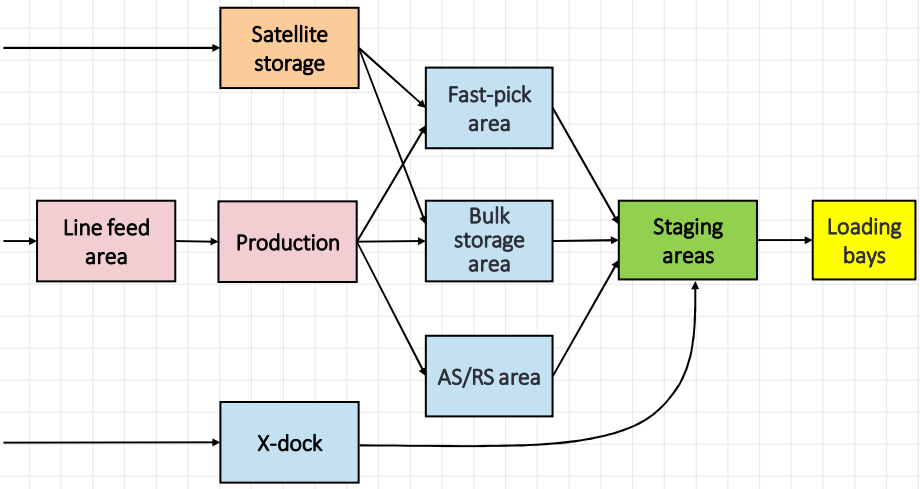
Loading yards

Warehousing application

In this case study we use bootstrapping and scenarios (BESO-SAA) by:

- Introducing a case study deriving from operation on warehouse outbound logistics.
- Formulating the problem as a stochastic optimization problem in a prescriptive analytics context
- Defining the predictive module on alternative forecasting techniques, including Maximum Entropy Bootstrap (with bagging)
- Using the bootstrap set as a seed for scenario-based optimization (BESO-SAA, *no DRO here*)

Internal flows

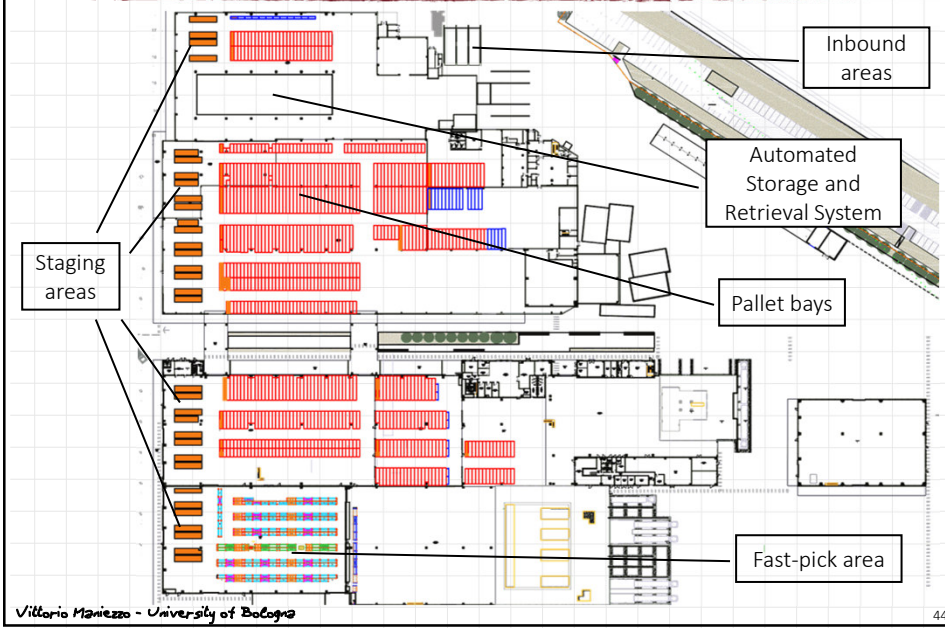


Staging area



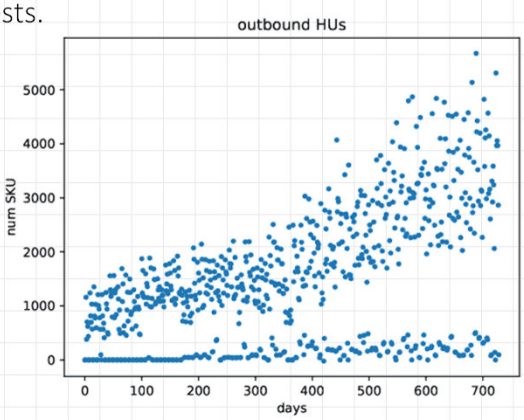
(Image from the web)

A warehouse



Daily number of outbound SKUs

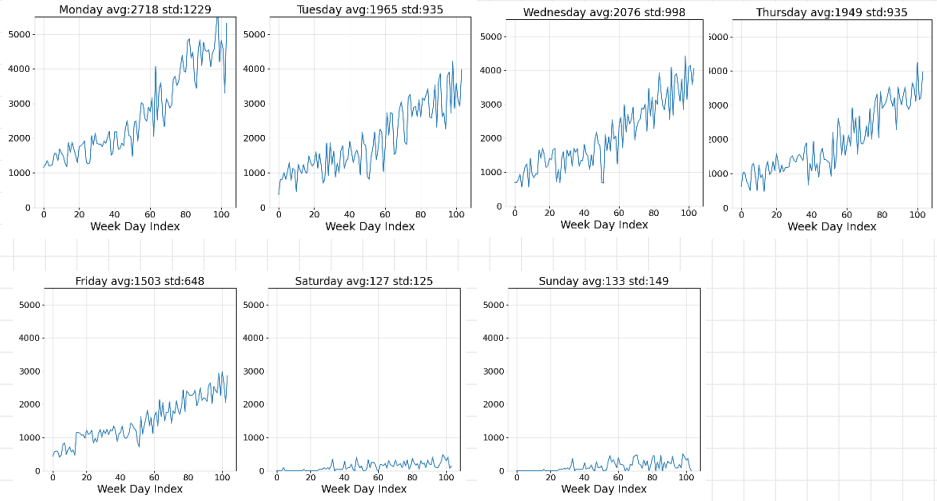
The problem was to determine if and when the current infrastructure would become insufficient to accommodate all outbound requests.



A series was available reporting the number of SKU (*pallets, HU*) and corresponding daily shipments (*trucks*) serviced at the warehouse.

Daily number of outbound SKUs

Separating the outbound requests for each day of the week.



If one has a hammer …



From global to detailed

Inputs available:

- The number of daily SKUs and corresponding trucks
- A customer (shipment) profile: ≈ 25 categories
- Request (load) distribution per category
- Distribution of pallets flowing from each WH area to staging areas for each category (mostly stable percentages)

Shipment forecast procedure

 # provides the set of shipments to serve and their load

1. Forecast total Monday shipments per category
2. Estimate each shipment's request from its category request distribution
3. Split requests across WH areas of origin
4. Compute pickup/loading time for each shipment/area pair

bootstrapping

We fit an AR(p) model on the prewhitened Mondays series (the most demanding) for each customer category

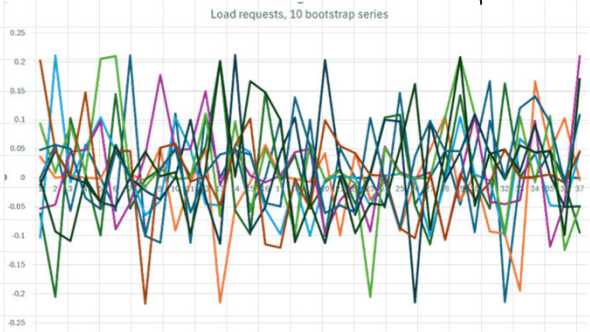
We extracted the residuals ε_t of the model.

We resampled the residuals ε_t as ε_t^* and simulated new series as:

$$x_t^* = \phi_1 x_{t-1}^* + \ \ldots + \phi_p x_{t-p}^* + \varepsilon_t^*$$

We bootstrapped the residual series and reconstructed request series.

Bagging applied to get demands ρ_{ij} in formulation F1



Deterministic scheduling: F1

Minimize the number of unserviced shipments:
It is a *GAP* if p_{ij}, *multiple knapsack* if p_i

J jobs (shipments, index j,k)

I machines (staging areas, index i)

p_{ij} processing time of job j on machine i (*pickup + load*)

Variables

x_{ij} job j assigned to machine i

ξ_j job j not assigned any machine

Standard scheduling nomenclature:
jobs = shipments,
machines = staging areas

Deterministic scheduling F1

Just minimize the number of unsupported shipments (trucks)

$$z(F1) = min \sum_{j \in J} \xi_j \quad (1)$$

$$subject\ to \quad \sum_{i \in M} x_{ij} + \xi_j = 1 \qquad j \in J \quad (2)$$

$$\sum_{j \in J} p_{ij} x_{ij} \leq t_{max} \qquad i \in I \quad (3)$$

$$x_{ij}, \xi_j \in \{0,1\} \qquad i \in I, j \in J \quad (4)$$

Maximize the number of on-time jobs on unrelated machines with a common due date. No sequencing decisions are needed

Stochastic scheduling, F1

$$z(F1) = min \sum_{j \in J} \xi_j \quad (1)$$

$$subject\ to \quad \sum_{i \in M} x_{ij} + \xi_j = 1 \qquad j \in J \quad (2)$$

$$\sum_{j \in J} \rho_{ij} x_{ij} \leq t_{max} \qquad i \in I \quad (3)$$

$$x_{ij}, \xi_j \in \{0,1\} \qquad i \in I, j \in J \quad (4)$$

Uncertainty on the load

Entropy tilting

For each scenario we optimized a F1 GAP-like instance with predefined costs for unserved customers

$$\mathrm{z(F1)} = min \quad \sum_{j\in J} w_j \xi_j \qquad (1)$$

$$subject\ to \quad \sum_{i\in M} x_{ij} + \xi_j = 1 \qquad j \in J \qquad (2)$$

$$\sum_{j\in J} \rho_{ij}^{\omega} x_{ij} \leq t_{max} \qquad i \in I \qquad (3)$$

$$x_{ij}, \xi_j \in \{0,1\} \qquad i \in I, j \in J \qquad (4)$$

The problem is NP-hard but very fast to solve with our dimensions.

$\mathrm{z(F1)}$ was then used as g_ω for scenario ω

Deterministic scheduling: F2

J jobs (shipments, index j,k)
I machines (areas, index i)
p_{ij} processing time of job j on machine i
r_j release time of job j

Variables

c_j completion time of job j
t_j tardiness of job j
s_j start time of job j
x_{ij} job j assigned to machine i
z_{jk} truck k follows truck j
t_{max} scheduling horizon

An unrelated-machine problem with a common due date. With job-level overtime, sequencing matters, the model needs pairwise precedence variables.
Jobs finishing before d cost nothing, the choice of jobs in the window [0, d] on each machine is a knapsack-like decision

Deterministic scheduling F2

Minimize the weighted total tardiness of the shipments

$$z(F2) = min \quad \sum_j w_j t_j \qquad (1)$$

$$subject\ to \quad \sum_{i \in M} x_{ij} = 1 \qquad j \in J \qquad (2)$$

$$z_{jk} + z_{jk} \leq 1 \qquad i \in I, j \in J \qquad (3)$$

$$z_{jk} + z_{kj} \geq x_{ij} + x_{ik} - 1 \qquad i \in I, j,k \in J, j<k \qquad (4)$$

$$c_j \leq s_k + M(3 - x_{ik} - x_{ij} - z_{jk}) \quad i \in I, j \neq k \in I \qquad (5)$$

$$c_j = s_j + \sum_i p_{ij} x_{ij} \qquad j \in J \qquad (6)$$

$$t_j \geq c_j - t_{max} \qquad j \in J \qquad (7)$$

$$s_j \geq r_j \qquad j \in J \qquad (*)$$

$$x_{ij}, z_{jk} \in \{0,1\} \qquad i \in I, j,k \in J \qquad (8)$$

$$c_j, t_j \geq 0 \qquad j \in J \qquad (9)$$

Stochastic scheduling F2

Same first-stage decisions across all scenarios

$$z(S2) = min \quad \sum_j w_j t_j \qquad (1)$$

$$subject\ to \quad \sum_{i \in M} x_{ij} = 1 \qquad j \in J \qquad (2)$$

$$z_{jk} + z_{kj} \leq 1 \qquad j,k \in I, j<k \qquad (3)$$

$$z_{jk} + z_{kj} \geq x_{ij} + x_{ik} - 1 \qquad i \in I, j,k \in J, j<k \qquad (4)$$

$$c_j \leq s_k + M(3 - x_{ik} - x_{ij} - z_{jk}) \quad i \in I, j \neq k \in I \qquad (5)$$

$$c_j = s_j + \sum_i \rho_{ij} x_{ij} \qquad j \in J \qquad (6)$$

$$t_j \geq c_j - t_{max} \qquad j \in J \qquad (7)$$

$$s_j \geq r_j \qquad j \in J \qquad (*)$$

$$x_{ij}, z_{jk} \in \{0,1\} \qquad i \in I, j,k \in J \qquad (8)$$

$$c_j, t_j \geq 0, \qquad j \in J \qquad (9)$$

BESO-SAA F2

We forecast these

$$z(S2) = min \quad \sum_{\omega\in\Omega} p_\omega \left(\sum_j w_j\, t_j^\omega\right) \qquad (1)$$

subject to

$$\sum_{i\in M} x_{ij} = 1 \qquad j \in J \qquad (2)$$

$$z_{jk} + z_{kj} \le 1 \qquad j,k \in I, j<k \qquad (3)$$

$$z_{jk} + z_{kj} \ge x_{ij} + x_{ik} - 1 \qquad i\in I, j,k \in J, j<k \qquad (4)$$

$$c_j^\omega \le s_k^\omega + M(3 - x_{ik} - x_{ij} - z_{jk}) \quad i\in I, j\neq k, \omega\in\Omega \qquad (5)$$

$$c_j^\omega = s_j^\omega + \sum_i \rho_{ij}^\omega x_{ij} \qquad j \in J, \omega \in \Omega \qquad (6)$$

$$t_j^\omega \ge c_j^\omega - t_{max} \qquad j \in J, \omega \in \Omega \qquad (7)$$

$$x_{ij}, z_{jk} \in \{0,1\} \qquad i \in I, j,k \in J \qquad (8)$$

$$c_j^\omega, t_j^\omega \ge 0, \; s_j^\omega \ge r_j \qquad j \in J, \omega \in \Omega \qquad (9)$$

Closer to the literature

For comparison, SAA based on formulation S3. Same load forecasts, no release time.
Ω is a set of S scenarios, with realizations ρ_{ij}^ω.
$x_{ijk} = 1$ if job j is in position k on machine i, $k \in K$.
c_{ik}^ω, completion time of position k, t_j^ω tardiness of job j in scenario ω .

$$z(F3) = min \quad \frac{1}{S}\sum_\omega \sum_i w_j t_j^\omega$$

subject to

$$\sum_i \sum_k x_{ijk} = 1 \qquad \forall j$$

$$\sum_j x_{ijk} \le 1 \qquad \forall i,k$$

$$\sum_j x_{ijk} \ge \sum_j x_{ijk+1} \qquad \forall i,k \text{ (no gaps)}$$

$$c_{ik}^\omega = c_{ik-1}^\omega + \sum_j \rho_{ij}^\omega x_{ijk} \qquad \forall i,k,\omega \; (c_{i0}^\omega = 0)$$

$$t_j^\omega \ge c_{ik}^\omega - t_{max} - M(1 - x_{ijk}) \qquad \forall i,j,k,\omega$$

$$x_{ijk} \in \{0,1\}, c_{ik}^\omega, t_j^\omega \ge 0$$

Smaller but looser. It was just used for validation on a limited number of test cases.

Optimization, details

Use case with 33 staging areas, no limit on forklift number, loading times computed on the actual warehouse corridors graph.

Forklift allowed to load only one pallet at a time. Travel speed the average of actual travel speeds.

Pickup locations defined according with ABC policy within each area.

Bootstrap on the Monday series of customer categories, tests on 75 and 125 bootstraps.

Implementation in C++, linking Cplex 22.11, max 600 sec CPU time.

What-if analysis with data projected 6 months ahead (point forecasts).

Results, optimization

We generated 75 and 125 bootstrapped series.

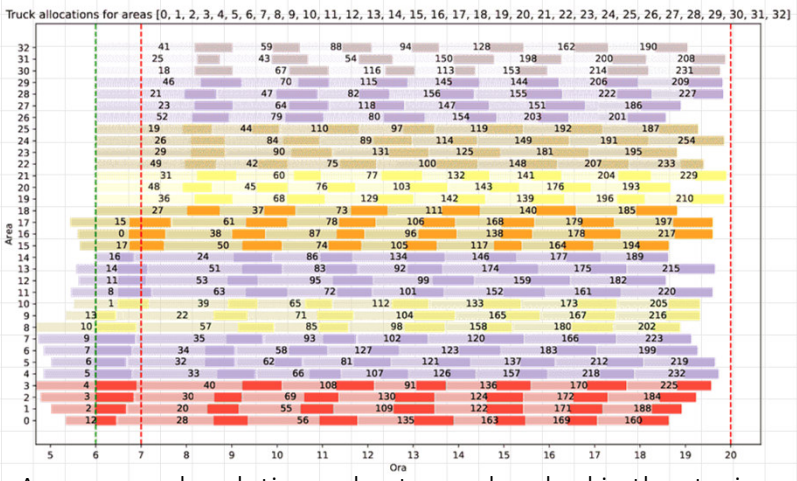
The cost increases significantly in the case of the 125 boostset.

- in the case of the 75 boostset 3 scenarios were deemed exceeding the future infrastructure.
- in the case of the 125 boostset 18 scenarios were deemed exceeding the future infrastructure.
- Solution was heuristic after 600 seconds, optimal guaranteed after 3600 (0% gap from previous incumbent).



✓ High quality solution obtained in limited computation time.
✓ The degree of robustness can be gauged using easy-to-understand parameters.

A result



A usage-ready solution, robustness absorbed in the staging preparation times

Comments

We applied the BESO approach to a real-world warehouse logistics case featuring:

- empirical probability distribution estimation enabled by bootstrapped forecasting
- data-driven, cost-aware scenario generation fed into stochastic SAA optimization
- easy application to real-world contexts even with nonstationary noisy historical data
- managerial interest also because of the justified insights it provides after easily described steps.

Takeaway: wherever decisions rest on uncertain forecasts, BESO turns that uncertainty into reliable decisions.