YEQT VIII, 3-5 November, 2014 Eindhoven

Tutorial:
System-Oriented Inventory Models for Spare Parts

Geert-Jan van Houtum

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Literature

Seminal paper:

- Sherbrooke, C.C., METRIC: A multi-echelon technique for recoverable item control, Operations Research 16, pp. 122-141, 1968

Two important books:

- Sherbrooke, C.C., Optimal *Inventory Modeling of Systems: Multi-Echelon Techniques*, Wiley, 1992. (2nd edition, Kluwer, 2004.)
- Muckstadt, J.A., *Analysis and Algorithms for Service Parts Supply Chains*, Springer, 2005.

This tutorial

Based on:

- Basten, R.J.I., and Van Houtum, G.J., “System-oriented inventory models for spare parts”, *Surveys in OR & MS 19*, pp. 34-55, 2014

In progress:

- Van Houtum, G.J., and Kranenburg, A.A., *Spare Parts Inventory Control under System Availability Constraints*, International Series in OR & MS, Springer.

CONTENTS

1. Introduction
2. Real-life networks
3. Single-location model with backordering
4. Single-location model with emergency shipments
5. METRIC model
6. Multi-location model with lateral and emergency shipments
7. Extensions
8. Applications in practice
9. Challenges for further research

1. Introduction

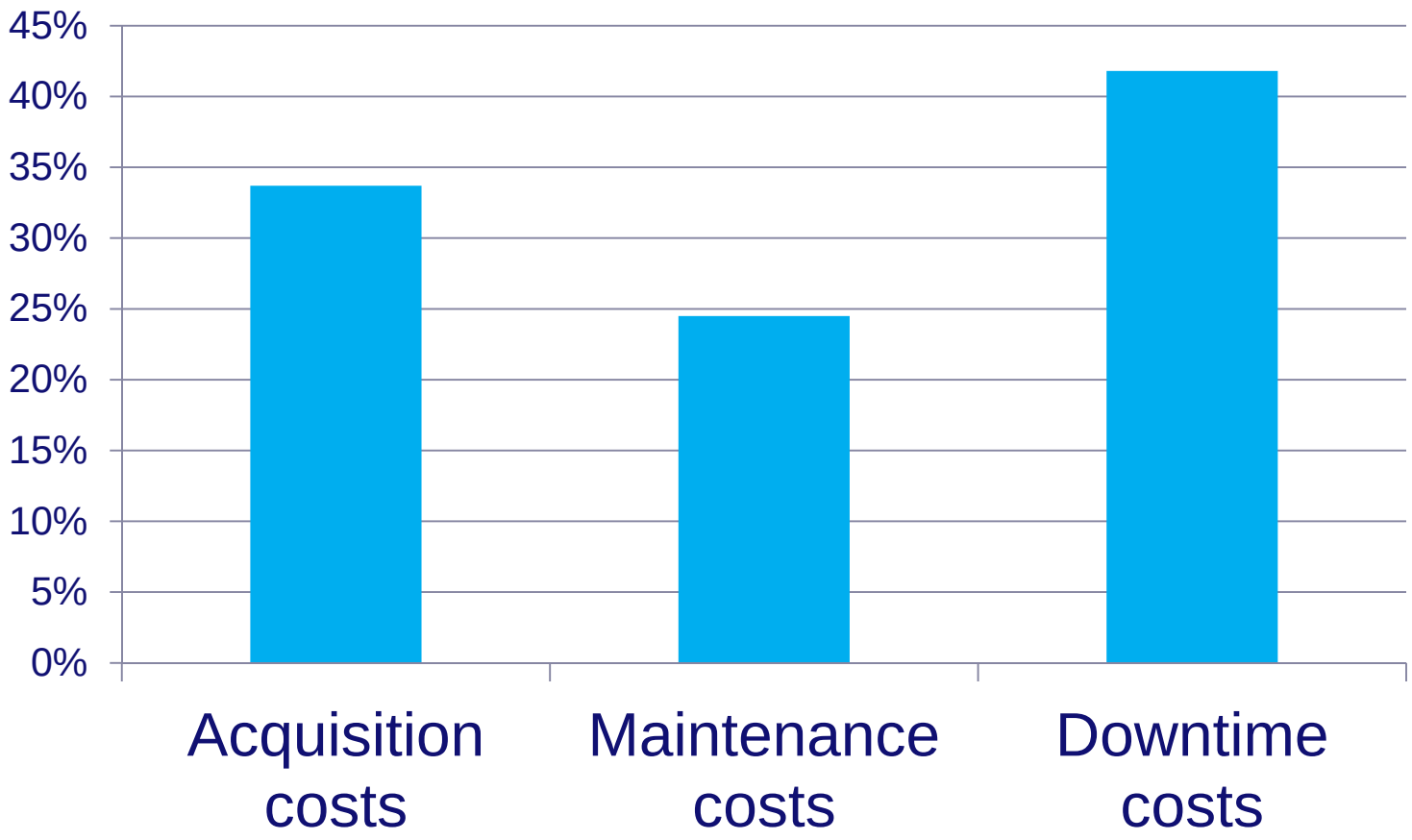
Total Cost of Ownership (TCO)

Total Cost of Ownership (TCO):

The total costs during the whole life cycle (perspective: user of the system)

Needs and requirements → Design → Production → Exploitation → Disposal

TCO for an example system



Size of maintenance industry

- Worldwide revenues after-sales services: 1500 billion US Dollars per year (AberdeenGroup [2003])
- Sales of spare parts and services in US: 8% of GNP (AberdeenGroup [2003])
- Manufacturers in US, Europe and Asia generate 26% of their revenues via services (Deloitte [2006])
- At an airline, 10% of all costs is constituted by maintenance costs (Lam [1995])

Long term trends

- Maintenance of complex systems becomes too complicated for users themselves
- Users require higher availabilities (less downtime)
- Users look at TCO

- Maintenance is outsourced to third party or OEM (pooling resources, pooling data, remote monitoring)
- More extreme: One sells function plus availability
- Feedback to design (better systems, higher sustainability)

Research topics

- **Spare parts management**
- Condition based maintenance
- Inventory models for spare parts and service tools
- Scheduling of service engineers
- Design of spare parts networks
- Forecasting of failures
- The effect of remote monitoring and diagnostics on total costs
- Service contracts and customer differentiation
- The effect of design decisions for new systems on their Total Cost of Ownership
- New business models for collaboration between users
- Game-theoretic models on the relationship between OEM-s, third parties and users
- …

Focus of this tutorial

Spare parts inventory models:

- for critical components
- of advanced capital goods
- with service level constraints for system-oriented service measures such as system availability or aggregate fill rate

Application in practice: Tactical planning level !

2. Real-life networks

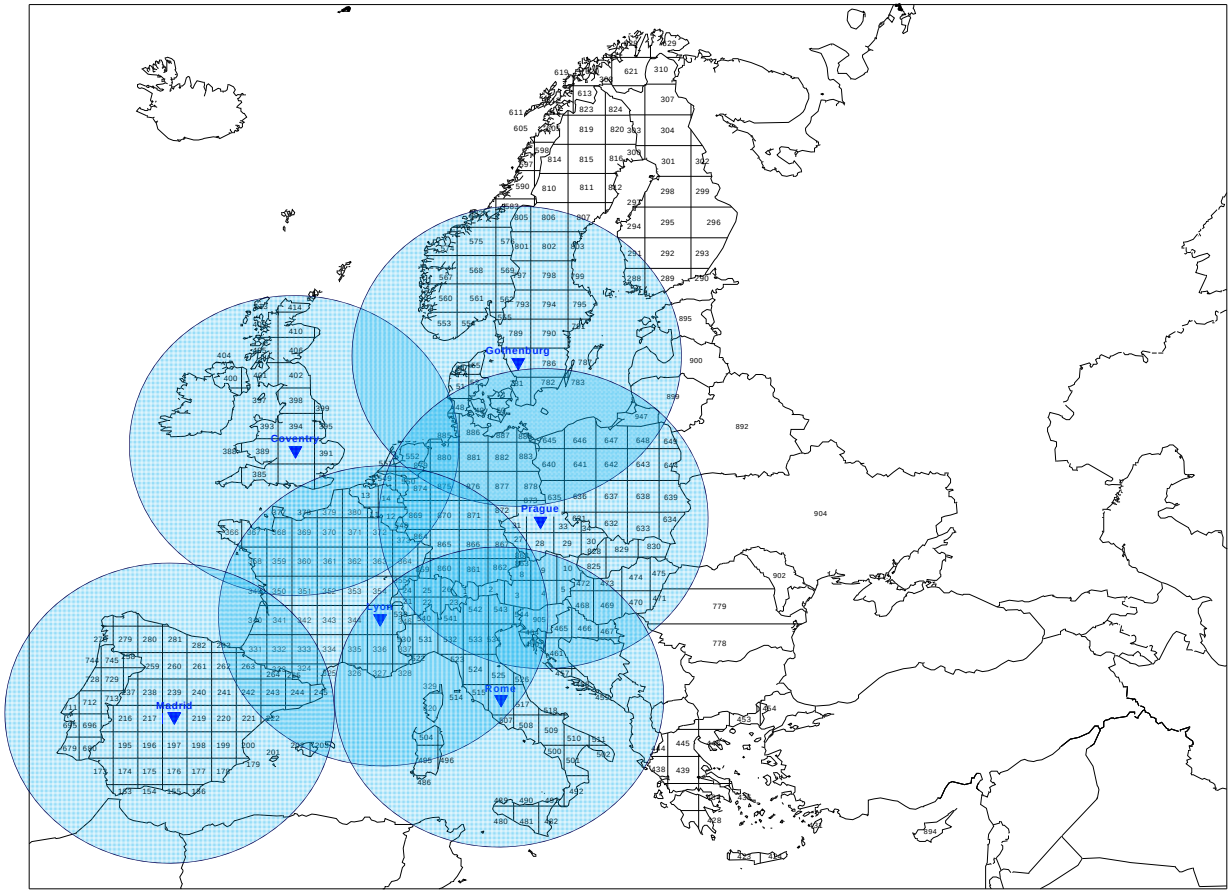
Network ASML

60 warehouses
5000 SKU's

Network ASML (cont.)

1. Normal delivery: 2 hrs.
2. Lateral transshipment: 14 hrs.
3. Emergency replenishment: 48 hrs.

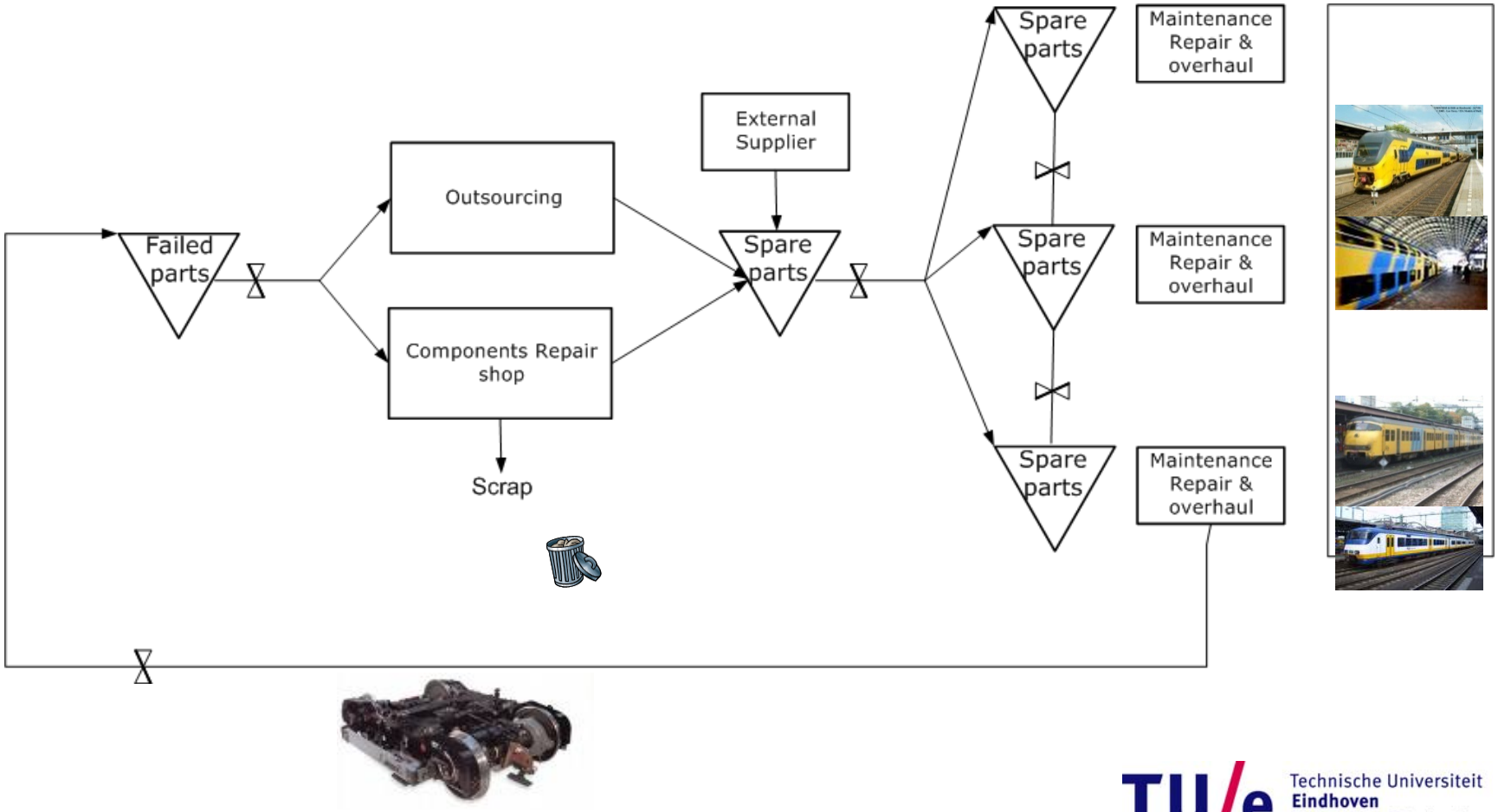
Network IBM (for next day deliveries)



scenario 1

time constraint

Network Nedtrain



User networks vs. OEM networks

User networks (typical for military systems)	OEM networks (typical for high-tech systems)
Preventive maintenance dominates	Corrective maintenance dominates
Two echelon levels in one region	Global network with two echelon levels
No emergency option	Emergency option at highest echelon level
Repairs at own repair shops	Repairs at original equipment manufacturers
Relatively loose service targets	Strict/high service targets

Examples of users maintaining their own system

User	Source
An electric utility company	[23]
Italian airports	[24]
KLM engineering & maintenance	Our experience (e.g., [25])
NedTrain	Our experience (e.g., [26])
Italian paper-making industry	[27]
Royal Netherlands Navy	[28,29,7]
US Air Force	[16,18,19]
US Coast Guard	[3]

Examples of OEMs maintaining sold system

OEM	Source
ASML	[1,34,35]
Cisco	Our experience (e.g., [36])
IBM	[37–39]
Océ	Our experience (e.g., [40,41])
Teradyne	[42]
Vanderlande Industries	Our experience (e.g., [43])
Volvo Parts Corporation	[44]

3. Single-location model with backordering

(cf. Chapter 2 of Sherbrooke [1992],

Single-item version: Feeney and Sherbrooke [1966])

3.1 Model description

3.2 Overview of assumptions

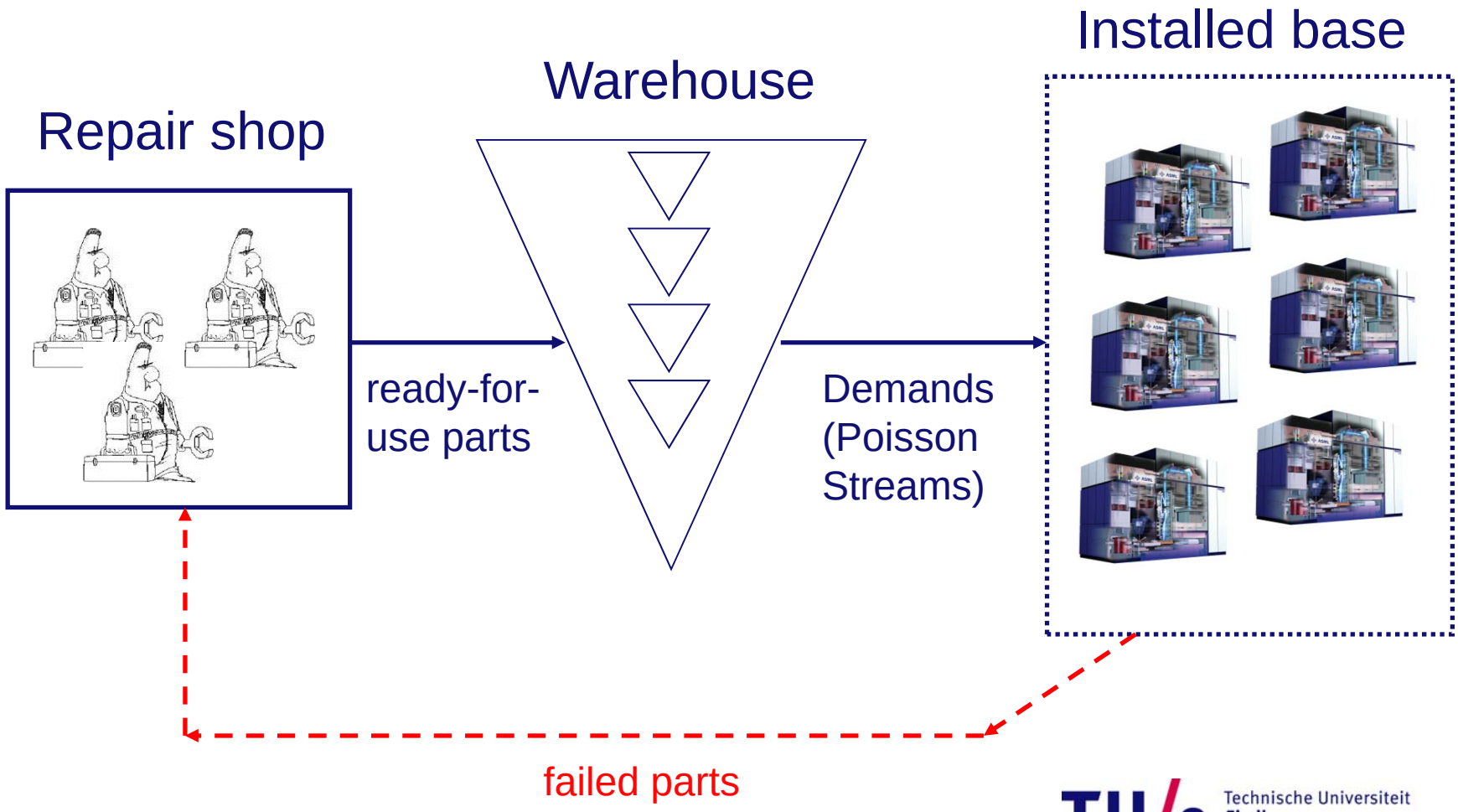
3.3 Evaluation

3.4 Optimization

3.5 Alternative optimization techniques and service measures

3.1. Model description

Model


Installed base
Warehouse
Repair shop
ready-for-use parts
Demands (Poisson Streams)
failed parts

Model (cont.)

- We use the terminology that is common for repairable LRU's (Line Replaceble Units). The model is also applicable for consumable LRU's.
- LRU's are denoted as SKU's (Stock-Keeping Units) in this presentation.
- We have an infinite time horizon.
- We look at the buy of the initial stock of spare parts for all SKU's.
- No emergency shipments.

Notation

Input variables:

- I : Set of SKU's, SKU's are numbered 1,..., $|I|$
- m_i : Demand rate for SKU i ($m_i \geq 0$)
- $M = \Sigma_{i \in I} m_i$: Total demand rate ($M > 0$)
- t_i : Mean repair leadtime for SKU i ($t_i > 0$)
- c_i^{h} : Price of a part of SKU i ($c_i^h > 0$)
- EBO^{obj} : Target level for aggregate mean number of backorders

Notation (cont.)

Decision variables:

- S_i : Basestock level for SKU i ($S_i \in \{0,1,...\}$)
- $\mathbf{S} = (S_1, S_2, ..., S_{|I|})$: Vector with all basestock levels denotes a solution

Notation (cont.)

Output variables:

- $C_i(S_i) = c_i^{\mathrm{h}} S_i$: Inventory holding costs for spare parts of SKU i
- $C(\mathbf{S}) = \Sigma_{i \in I} C_i(S_i) = \Sigma_{i \in I} c_i^{\mathrm{h}} S_i$: Total inventory holding costs
- $EBO_i(S_i)$: Mean number of backorders of SKU i
- $EBO(\mathbf{S}) = \Sigma_{i \in I} EBO_i(S_i)$: Aggregate mean number of backorders

Optimization problem

Problem (P)*:*

$$\min \quad C(\mathbf{S})$$

subject to

$$EBO(\mathbf{S}) \leq EBO^{\text{obj}}$$

$$S_i \in \{0,1,\ldots\} \quad \text{for all } i \in I$$

Relation with availability:

Availability($\mathbf{S}$) ≈ 1 – ($EBO(\mathbf{S})$ / N)

where N is the number of machines

3.2. Overview of assumptions

“ 1. Demands for the different SKU’s occur according to independent Poisson processes ”

- A failure of a component does not lead to additional failures of other components
- Assumption of Poisson demand processes is justified when:
  - lifetimes of components are exponential, or
  - lifetimes are generally distributed and number of machines is sufficiently large (a merge of many renewal processes gives a process that is close to Poisson)

“ 2. For each SKU, the demand rate is constant ”

- Justified in case fraction of machines that is down, is always sufficiently small:
  - either because downtimes are short in general
  - or downtimes occur only rarely

“ 3. Repair leadtimes for different SKU’s are independent and repair leadtimes of the same SKU are i.i.d. ”

- See repair leadtimes as <u>planned repair leadtimes</u>

“ 4. A one-for-one replenishment strategy is applied for all SKU’s ”

- Justified in case:
  - there are no fixed ordering costs at all, or
  - fixed ordering costs are small relative to the prices of the SKU’s

3.3. Evaluation

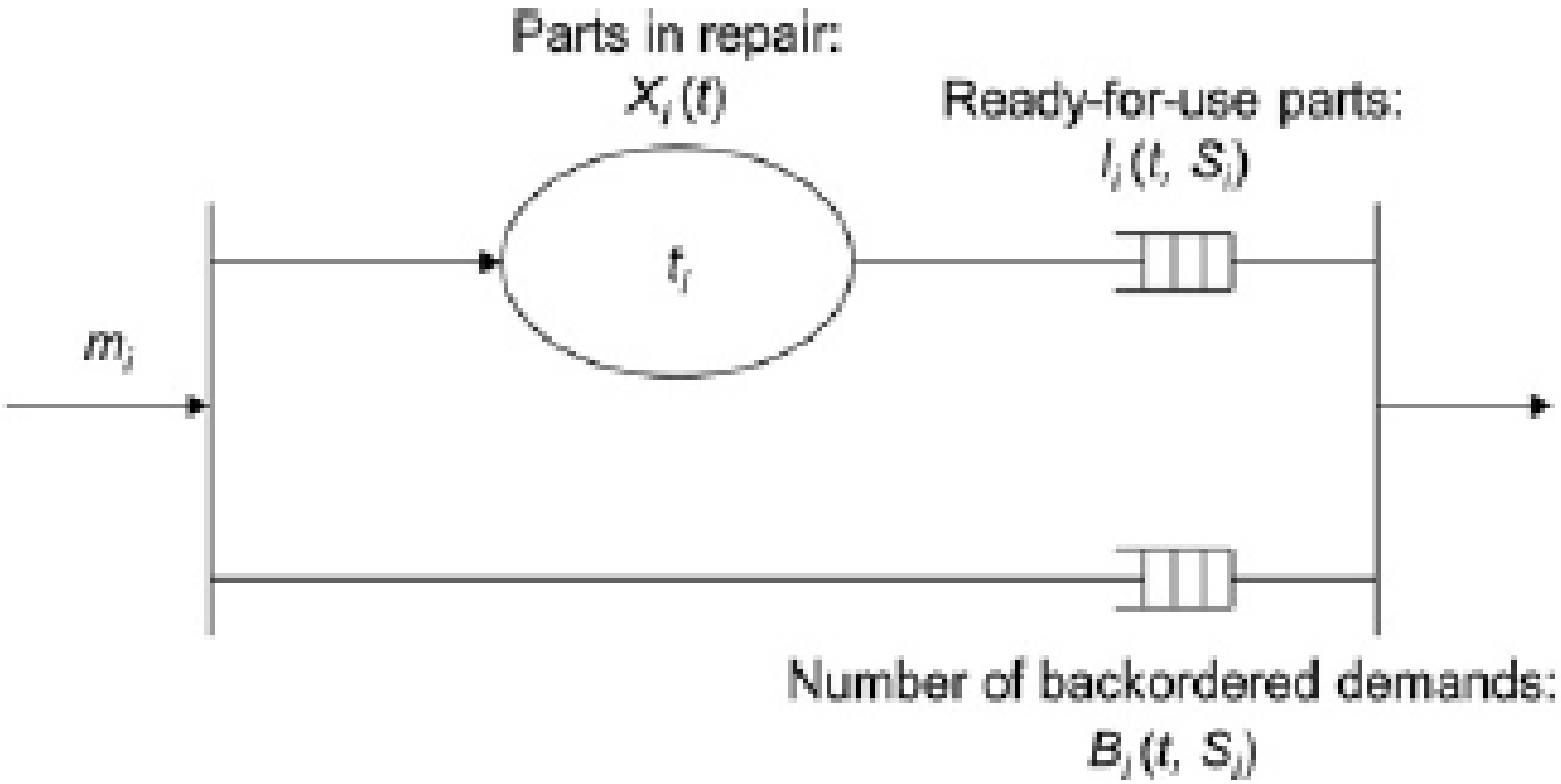
Extra notation

Evaluation can be done per SKU

<u>Extra notation per SKU i :</u>

- $X_i(t)$: number of parts in repair at time t
- $I_i(t,S_i)$: number of parts on hand at time t
- $B_i(t,S_i)$: number of backordered demands at time t
- X_i , $I_i(t,S_i)$, $B_i(t,S_i)$: corresponding steady-state variables

Petri net of repair and demand fulfilment process



Possible states

States that can occur for $(X_i(t), I_i(t,S_i), B_i(t,S_i))$:

(0, S_i, 0) : Nothing in repair, S_i good parts on stock

(1, S_i-1, 0): 1 part in repair, S_i-1 good parts on stock

...

(S_i-1, 1, 0): S_i-1 parts in repair, 1 good part on stock

(S_i, 0, 0) : S_i parts in repair, no parts on stock anymore

(S_i+1, 0,1): S_i+1 in repair, no stock, 1 request in backlog

...

Equations

- $I_i(t,S_i) = (S_i - X_i(t))^+$
- $B_i(t,S_i) = (X_i(t) - S_i)^+$
- Stock balance equation:

$$X_i(t) + I_i(t,S_i) - B_i(t,S_i) = S_i$$

And thus also:

- $I_i(S_i) = (S_i - X_i)^+$
- $B_i(S_i) = (X_i - S_i)^+$
- $X_i + I_i(S_i) - B_i(S_i) = S_i$

Palm's Theorem

Palm's Theorem (cf. Palm [1938]):

If at a certain unit the arrival process of jobs is Poisson with rate λ and if the leadtimes for the jobs are independent and identically distributed random variables corresponding to any distribution with mean EW, then the steady state probability distribution for the number of jobs present in that unit is a Poisson distribution with mean $\lambda\, EW$.

- Developed for an M|G|∞ queue
- Result is easily seen for deterministic leadtimes
- Generalizes Little's law for this particular system

Application of Palm's Theorem

Lemma 3.1. *Let $i \in I$.*

(i) *The pipeline X_i is Poisson distributed with mean $m_i t_i$, i.e.:*

$$\mathbb{P}\{X_i = x\} = \frac{(m_i t_i)^x}{x!} e^{-m_i t_i}, \quad \forall x \in \mathbb{N}_0.$$

(ii) *The distribution of the stock on hand $I_i(S_i)$ is given by:*

$$\mathbb{P}\{I_i(S_i) = x\} = \begin{cases} \sum_{y=S_i}^{\infty} \mathbb{P}\{X_i = y\} & \text{if } x = 0; \\ \mathbb{P}\{X_i = S_i - x\} & \text{if } x \in \{1, \ldots, S_i\}. \end{cases}$$

(iii) *The distribution of the number of backordered demands $B_i(S_i)$ is given by:*

$$\mathbb{P}\{B_i(S_i) = x\} = \begin{cases} \sum_{y=0}^{S_i} \mathbb{P}\{X_i = y\} & \text{if } x = 0; \\ \mathbb{P}\{X_i = x + S_i\} & \text{if } x \in \mathbb{N}. \end{cases}$$

Last step

$$EBO_i(S_i) = \mathbb{E}B_i(S_i) = \sum_{x=S_i+1}^{\infty} (x - S_i)\mathbb{P}\{X_i = x\}$$

$$= m_i t_i - S_i + \sum_{x=0}^{S_i} (S_i - x)\mathbb{P}\{X_i = x\}, \quad \forall S_i \in \mathbb{N}_0.$$

3.4. Optimization

Alternative problem

Problem (P)*:*

$$\min \quad C(\mathbf{S})$$

subject to

$$EBO(\mathbf{S}) \leq EBO^{\text{obj}}$$

$$S_i \in \{0,1,\ldots\} \quad \text{for all } i \in I$$

Kind of knapsack problem; hard to solve

Alternative problem (cont.)

Problem (Q):

$$\min \quad C(\mathbf{S}) = \Sigma_{i \in I}\, c_i^{\mathrm{h}}\, S_i$$

$$\min \quad EBO(\mathbf{S}) = \Sigma_{i \in I}\, EBO_i(S_i)$$

subject to

$$\mathbf{S} \in \{(S_1, S_2, \ldots, S_{|I|}) \mid S_i \in \{0,1,\ldots\} \text{ for all } i\}$$

$\Rightarrow$ ***Multi-objective programming problem***

We derive so-called efficient solutions

Example 1

Input variables for Problem (Q)*:*

- $|I| = 3$
- $m_1 = 15$, $m_2 = 5$, $m_3 = 1$, $M = 21$ demands/yr
- $t_1 = t_2 = t_3 = 1/6$ yrs
- $c_1^{\mathrm{h}} =$ € 1,000, $c_2^{\mathrm{h}} =$ € 3,000, $c_3^{\mathrm{h}} =$ € 20,000

Extra input variable for Problem (P)*:*

- $EBO^{\mathrm{obj}} = 0.1$

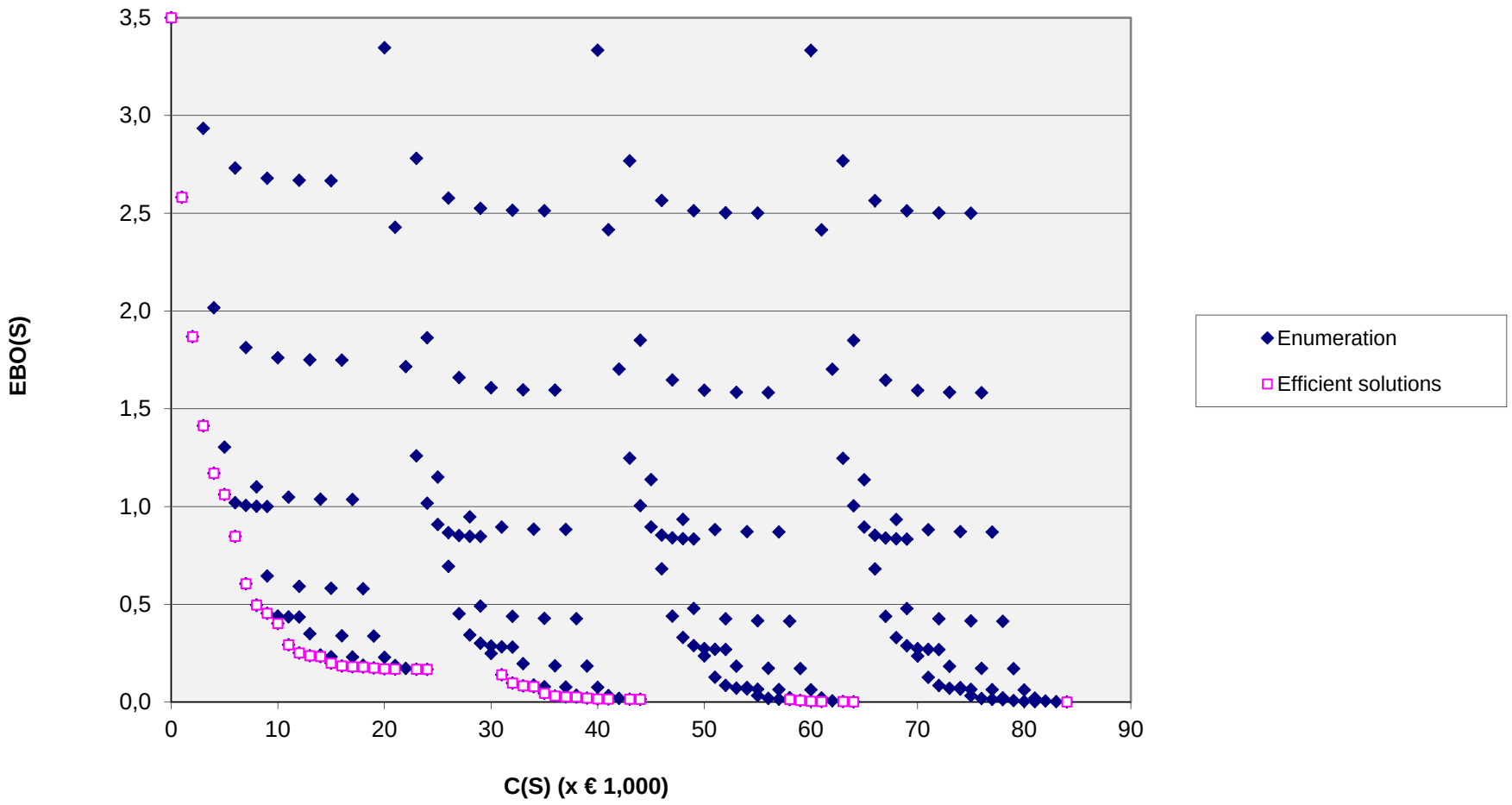
Example 1 (cont.)

Enumeration exercise:

- Consider plausible values for the basestock levels
- Compute $C(\mathbf{S})$
- Compute $EBO(\mathbf{S})$
- Plot them in a $C(\mathbf{S})$
- vs. $EBO(\mathbf{S})$ figure

S_{1}	S_{2}	S_{3}	C(S) (€)	EBO(S)
0	0	0	-	3,500
1	0	0	1.000	2,582
2	0	0	2.000	1,869
3	0	0	3.000	1,413
4	0	0	4.000	1,171
5	0	0	5.000	1,062
6	0	0	6.000	1,020
7	0	0	7.000	1,006
8	0	0	8.000	1,002
9	0	0	9.000	1,000
0	1	0	3.000	2,935
1	1	0	4.000	2,017
2	1	0	5.000	1,304
3	1	0	6.000	0,848
4	1	0	7.000	0,605
5	1	0	8.000	0,497
6	1	0	9.000	0,455
7	1	0	10.000	0,440
8	1	0	11.000	0,436
9	1	0		
0	2	0		

Example 1 (cont.)



Example 1 (cont.)

For Problem (Q)*, we obtain:*

- Efficient solutions
- The whole efficient frontier
- Observation: ***An optimal solution for Problem (P) is efficient for Problem (Q), and vice versa***

Optimal solution for Problem (P)*:*

- $\mathbf{S} = (6, 2, 1)$
- $C(\mathbf{S}) =$ € 32,000
- $EBO(\mathbf{S}) = 0.098$

Alternative problem (cont.)

Problem (Q):

$$\min \quad C(\mathbf{S}) = \Sigma_{i \in I}\, c_i^{\mathrm{a}}\, S_i$$

Separable

$$\min \quad EBO(\mathbf{S}) = \Sigma_{i \in I}\, EBO_i(S_i)$$

Separable

subject to

$$\mathbf{S} \in \{(S_1, S_2, \ldots, S_{|I|}) \mid S_i \in \{0,1,\ldots\} \text{ for all } i\}$$

Cartesian product

=> Problem (Q) is separable (cf. Fox, 1966)

=> Efficient solutions via greedy algorithm

Greedy algorithm

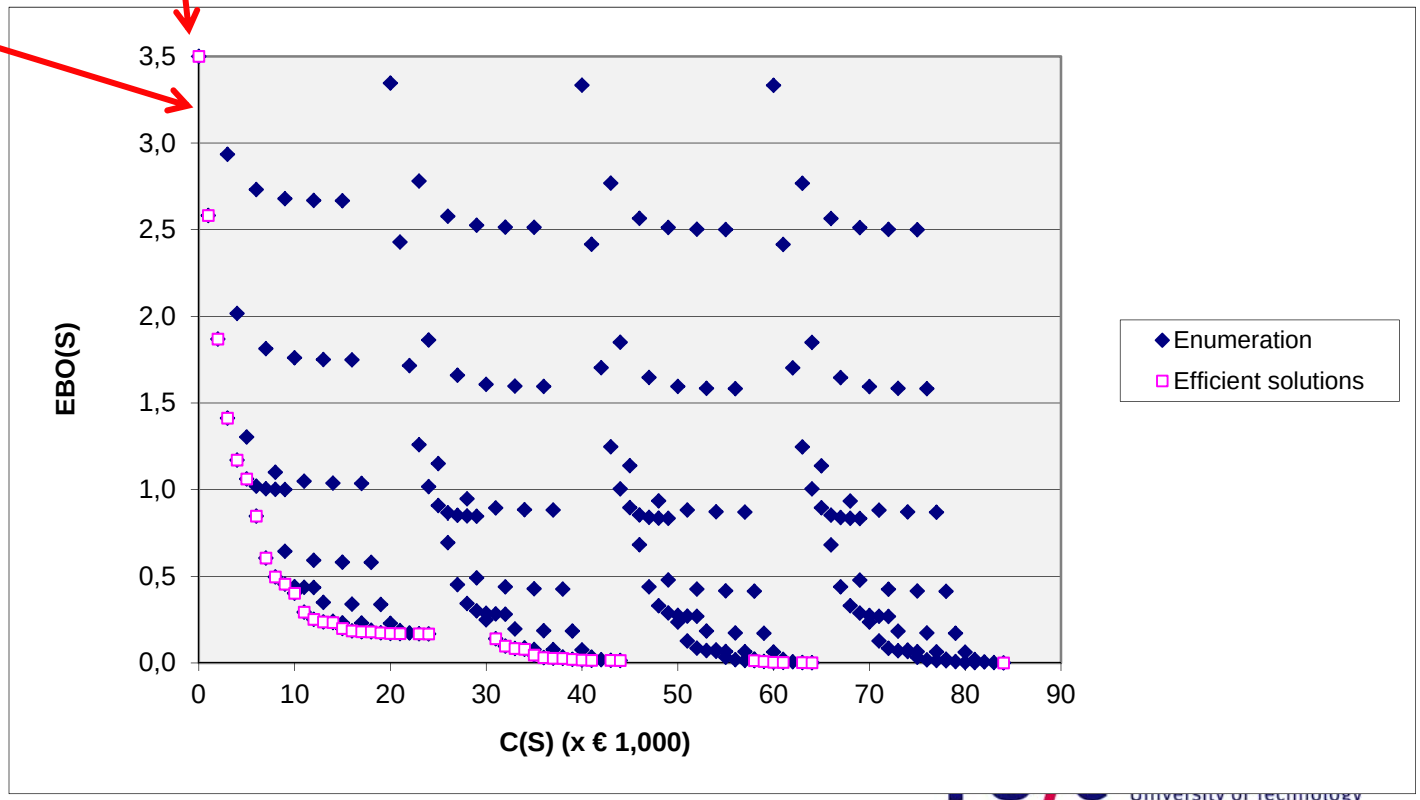
An obvious efficient solution:

- $\mathbf{S} = \{(0, 0, \ldots, 0)$
- $C(\mathbf{S}) = 0$ => All other solutions: Higher costs
- $EBO(\mathbf{S}) = \Sigma_{i \in I} EBO_i(0) = \Sigma_{i \in I} m_i t_i$

Greedy algorithm (cont.)

Starting point

Next:
Look for the steepest decrease in *EBO*(**S**) against the lowest increase in *C*(**S**)



Greedy algorithm (cont.)

<u>Generation of a next efficient solution:</u>

- Current solution: $\mathbf{S}$
- If S_i would be increased with 1 unit:
  - Decrease for $EBO(\mathbf{S}) = \Delta_i\, EBO(\mathbf{S}) = \Delta\, EBO_i(S_i) = \ldots$ (see proof of Lemma 3.2)
  - Increase in $C(\mathbf{S}) = \Delta_i\, C(\mathbf{S}) = c_i^{\mathrm{h}}$
  - $\Gamma_i := \Delta_i\, EBO(\mathbf{S}) / \Delta_i\, C(\mathbf{S})$
- Pick the SKU with the largest Γ_i ('biggest bang for the buck')

Greedy algorithm (cont.)

Set of efficient solutions

Algorithm 3.1 (*Greedy Algorithm*).

Step 1. $S_i := 0$ for all $i \in I$ (so $\mathbf{S} := (0, \ldots, 0)$);
$\mathcal{E} := \{\mathbf{S}\}$;
$C(\mathbf{S}) := 0$ and $EBO(\mathbf{S}) := \sum_{i \in I} m_i t_i$.

Step 2. $\Gamma_i := \frac{1}{c_i^h}(1 - \sum_{x=0}^{S_i} \mathbb{P}\{X_i = x\})$ for all $i \in I$;
$k := \arg\max_{i \in I} \Gamma_i$;
$\mathbf{S} := \mathbf{S} + e_k$;
$\mathcal{E} := \mathcal{E} \cup \{\mathbf{S}\}$.

Step 3. $C(\mathbf{S}) := C(\mathbf{S}) + c_k^h$;
$EBO(\mathbf{S}) := EBO(\mathbf{S}) - 1 + \sum_{x=0}^{S_k} \mathbb{P}\{X_k = x\}$;
If 'stop criterium', then stop, else go to step 2.

Greedy algorithm (cont.)

Lemma 3.3

Algorithm 3.1 generates efficient solutions for Problem (Q).

Proof: Via Fox (1966).

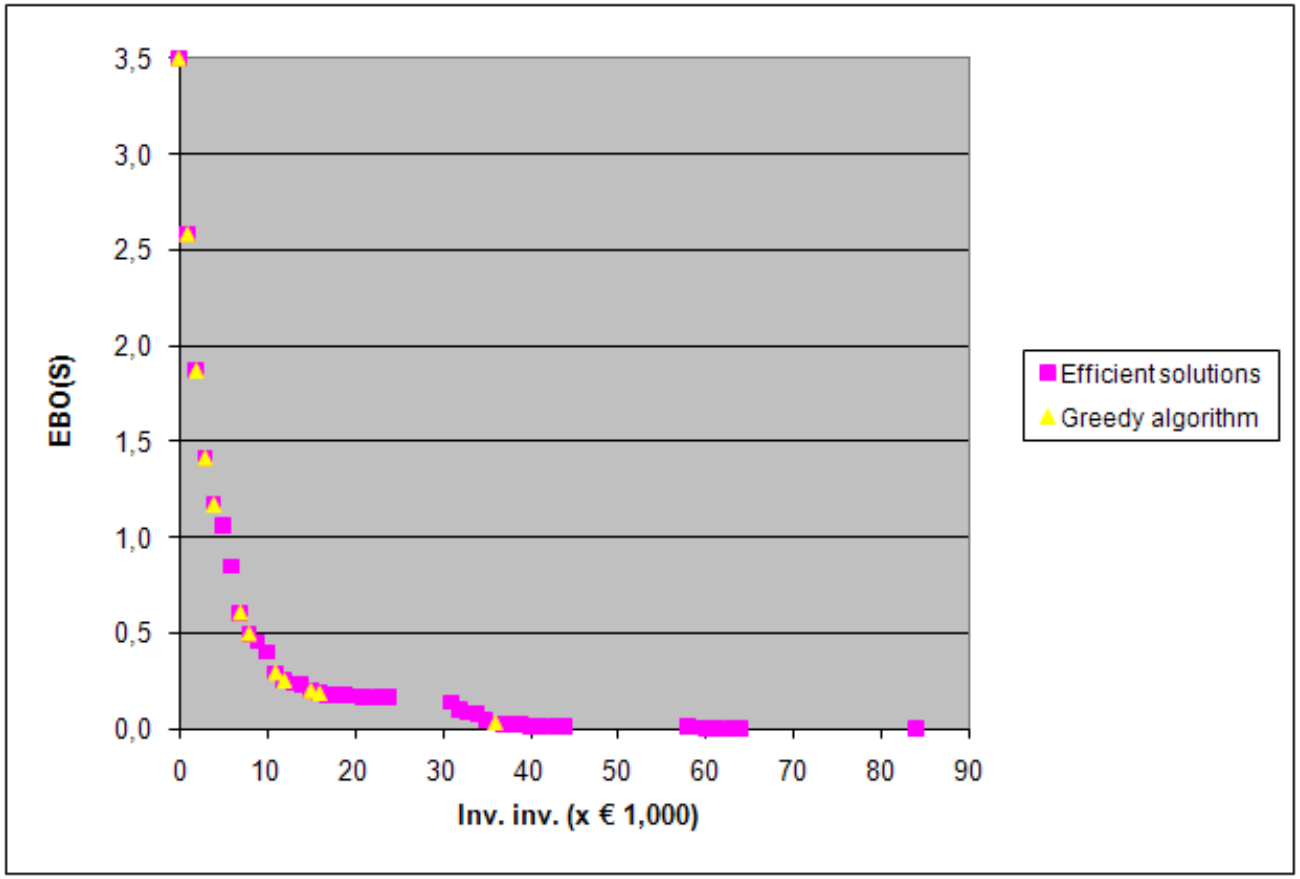
Alternative: Directly via the logic that always the most negative slope is chosen for the curve formed by the generated solutions.

Example 1 (cont.)

Application of greedy algorithm:

iteration	Γ_1	Γ_2	Γ_3	k	S_1	S_2	S_3	$EBO(\mathbf{S})$	$C(\mathbf{S})$ (Euros)
0	–	–	–	-	0	0	0	3.500	0
1	$9.18 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	1	1	0	0	2.582	1,000
2	$7.13 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	1	2	0	0	1.869	2,000
3	$4.56 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	1	3	0	0	1.413	3,000
4	$2.42 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	1	4	0	0	1.171	4,000
5	$1.09 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	2	4	1	0	0.605	7,000
6	$1.09 \cdot 10^{-4}$	$6.77 \cdot 10^{-5}$	$7.68 \cdot 10^{-6}$	1	5	1	0	0.497	8,000
7	$4.20 \cdot 10^{-5}$	$6.77 \cdot 10^{-5}$	$7.68 \cdot 10^{-6}$	2	5	2	0	0.293	11,000
8	$4.20 \cdot 10^{-5}$	$1.74 \cdot 10^{-5}$	$7.68 \cdot 10^{-6}$	1	6	2	0	0.251	12,000
9	$1.42 \cdot 10^{-5}$	$1.74 \cdot 10^{-5}$	$7.68 \cdot 10^{-6}$	2	6	3	0	0.199	15,000
10	$1.42 \cdot 10^{-5}$	$3.47 \cdot 10^{-6}$	$7.68 \cdot 10^{-6}$	1	7	3	0	0.185	16,000
11	$4.25 \cdot 10^{-6}$	$3.47 \cdot 10^{-6}$	$7.68 \cdot 10^{-6}$	3	7	3	1	0.031	36,000

Example 1 (cont.)



We obtain a subset of all efficient solutions !

Example 1 (cont.)

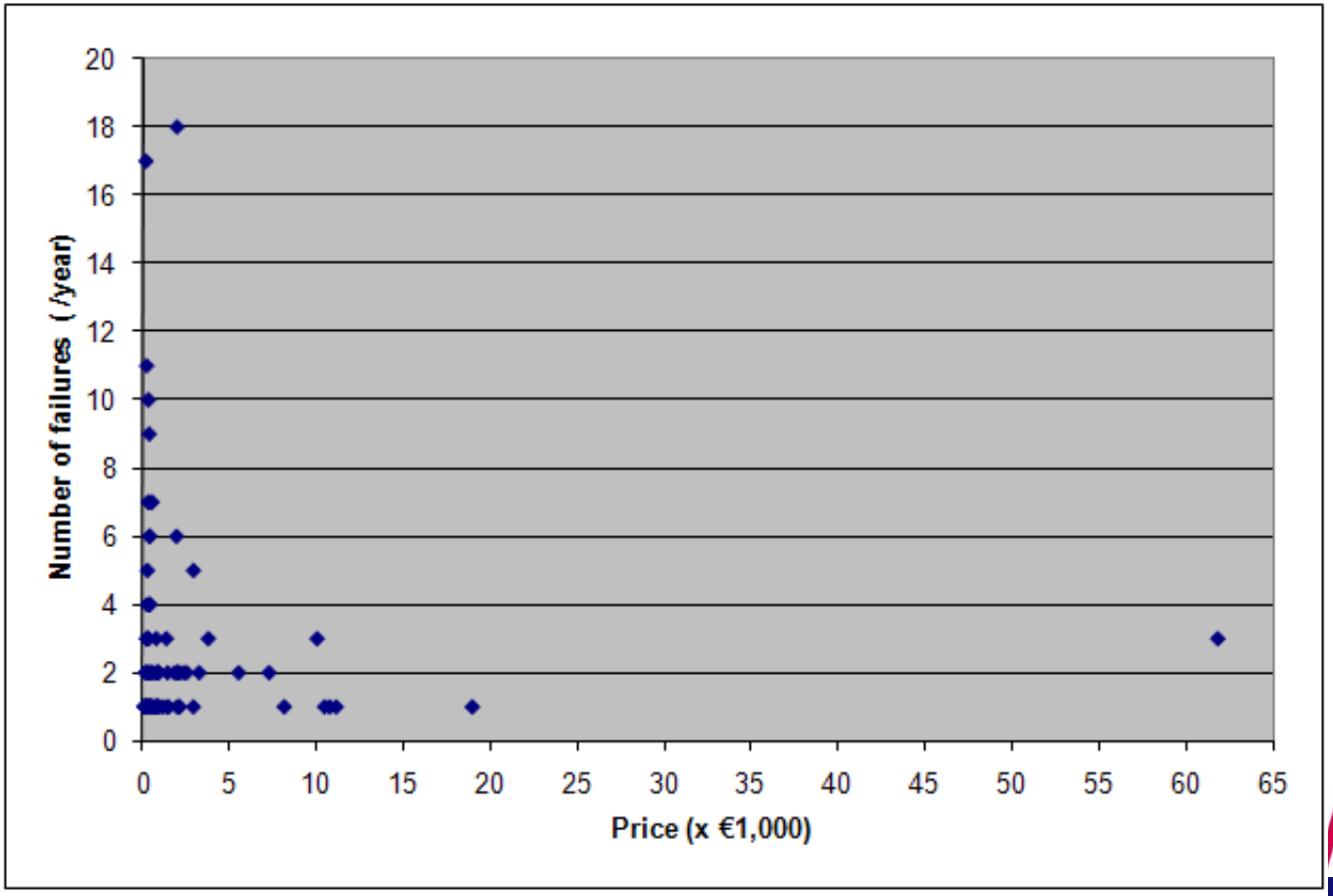
Via greedy algorithm:

- Heuristic solution for Problem (P) with $EBO_{obj} = 0.1$:
  - $\mathbf{S} = (7, 3, 1)$, $C(\mathbf{S}) =$ € 36,000, $EBO(\mathbf{S}) = 0.031$
  - Optimality gap = (36000-33000)/33000 = 9.1%

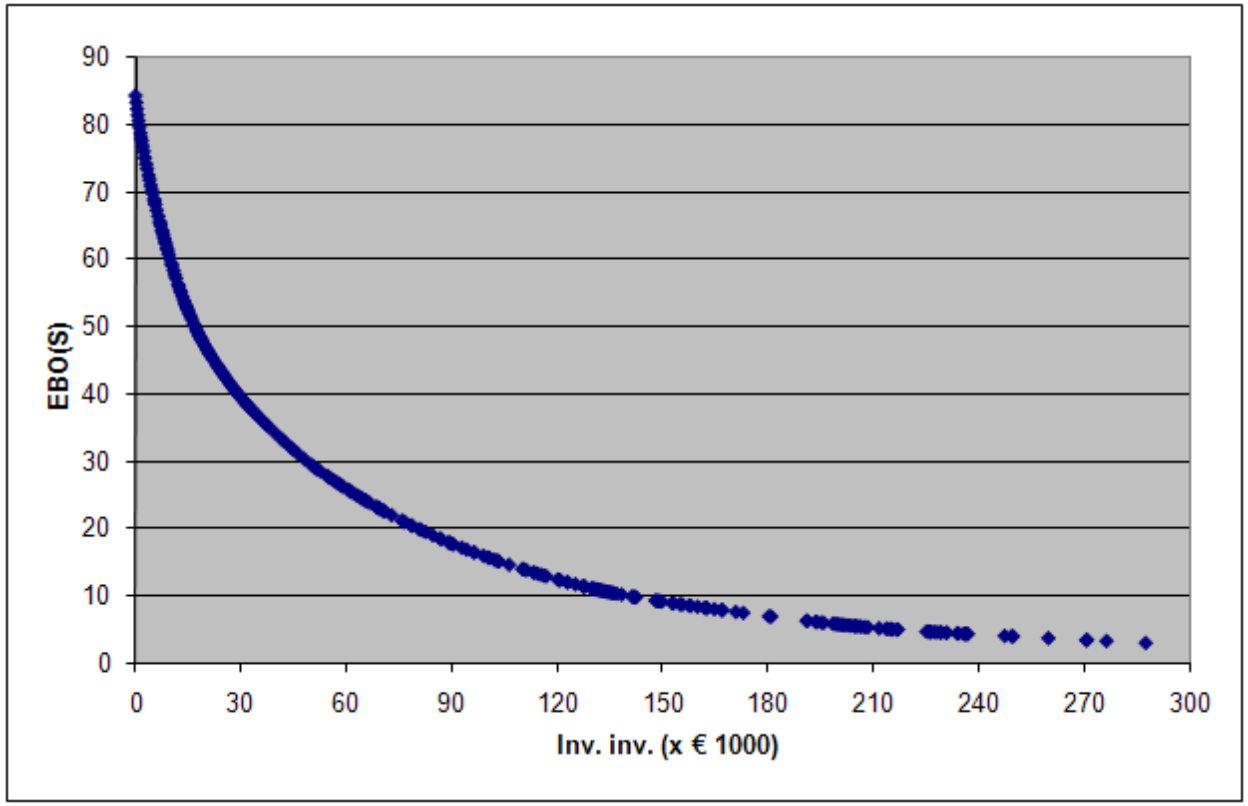
- Optimal solution for Problem (P) with $EBO_{obj} = 0.031$:
  - $\mathbf{S} = (7, 3, 1)$, $C(\mathbf{S}) =$ € 36,000, $EBO(\mathbf{S}) = 0.031$

Example 2

Real-life data, 99 SKU's



Example 2 (cont.)



=> Smooth line, good heuristic solutions for Problem (P)

Conclusions w.r.t. greedy algorithm

- Generates efficient solutions for Problem (Q)
- Does not generate all efficient solutions, but a certain subset of solutions
- This subset has a certain robustness: small changes in the input parameters (e.g., demand rates!) lead to small changes in the subset of generated solutions
- Leads to a heuristic solution for Problem (P). Conjecture: This solution is robust. (Notice: This does not hold for the exact solution of Problem (P).)

Conclusions w.r.t. greedy algorithm (cont.)

- Heuristic solution is optimal for Problem (P) for some specific values of EBO^{obj}
- Generally, the heuristic solution for Problem (P) will be good when one has many SKU's
- Simple and easy to understand
- Easy to implement in practice
- Requires little computational effort

3.5. Alternative optimization techniques and service measures

Alternative techniques:

- Langrange relaxation
- Dantzig-Wolfe decomposition

-> Both give the same efficient solutions as greedy algorithm

Alternative service measures:

- Aggregate mean waiting time
- Aggregate fill rate
- …

-> Works as long as you have the required convexity proporties

Why is robustness of the heuristic solutions important?

<u>Spare parts planning in practice:</u>

- Executed every 3 months, say
- Per planning moment:
  - Generation of new forecasts for the demand rates of all SKU's
  - Application of the greedy heuristic
- Implementation of new solution:
  - Both increasing and decreasing the stock of a SKU gives some costs

Open problems

- Show/prove that the greedy heuristic leads to robust solutions
- Similarly for Dantzig-Wolfe decomposition (may be easier when we go to dimension 3 or higher)
- How to use of the model in a rolling horizon setting such that basestock levels do not change too much

4. Single-location model with emergency shipments

Emergency Shipments

- Assumption for basic model:

  If a demand cannot be immediately fulfilled from stock, then the demand is backordered
- In several practical situations:
  - Downtimes of machines are very expensive
  - In case of a stockout, a demand will be satisfied in an alternative way, i.e., via a fast repair procedure or via an emergency shipment
  - 'lost sales' instead of 'backordering'

Changes in model assumptions

- t_i^{em} : Average time for an emergency shipment for SKU i
- c_i^{em} : Cost of an emergency shipment for SKU i, minus cost of a normal repair ($c_i^{em} \geq 0$)
- c_i^{h} : Inventory holding cost per time unit per part of SKU i ($c_i^{\mathrm{h}} > 0$)
- $W_i(S_i)$: Mean waiting time for a demand for SKU i
- $W(\mathbf{S}) = \Sigma_{i \in I} (m_i/M)\ W_i(S_i)$: Aggregate mean waiting time for an arbitrary demand for all SKU's together

Changes in model assumptions (cont.)

- W^{obj} : Target level for $W(\mathbf{S})$
- Problem formulation:

$$
\begin{aligned}
(\text{P''}) \qquad & \min \qquad \hat{C}(\mathbf{S}) \\
& \text{subject to } W(\mathbf{S}) \leq W^{\text{obj}}, \\
& \mathbf{S} \in \mathscr{S}.
\end{aligned}
$$

- Link with availability:

$$A(\mathbf{S}) \approx 1 - \frac{MW(S)}{Z}.$$

Evaluation

- Total costs: $\hat{C}(\mathbf{S}) = \sum_{i \in I} \hat{C}_i(S_i).$

$$\hat{C}_i(S_i) = c_i^{\mathrm{h}} S_i + m_i(1 - \beta_i(S_i))c_i^{\mathrm{em}},$$

- Formula for $W(\mathbf{S})$: $W(\mathbf{S}) = \sum_{i \in I} \frac{m_i}{M} W_i(S_i).$

$$W_i(S_i) = (1 - \beta_i(S_i))t_i^{\mathrm{em}},$$

- Expression for fill rate $\beta_i(S_i)$ (steady-state behavior is equivalent to behavior in Erlang loss system):

$$\beta_i(S_i) = 1 - \underbrace{\frac{\frac{1}{S_i!}\rho_i^{S_i}}{\sum_{j=0}^{S_i} \frac{1}{j!}\rho_i^j}}_{\text{Erlang loss probability}}, \qquad \rho_i := m_i t_i.$$

Optimization

- Switch to related multi-objective problem:

$$
\begin{array}{lll}
(\mathrm{Q}'') & \min & \hat{C}(\mathbf{S}) \\
 & \min & W(\mathbf{S}) \\
 & \text{subject to } \mathbf{S} \in \mathscr{S}. &
\end{array}
$$

- Karush (1957): Erlang loss probability is strictly convex and decreasing as a function of the number of servers

Optimization (cont.)

- Hence:
  - $\beta_i(S_i)$ is strictly concave and increasing
  - $W_i(S_i)$ is strictly convex and decreasing
  - $\hat{C}_i(S_i)$ is convex
  - $\hat{C}_i(S_i)$ may now be decreasing for smaller values of S_i
- Define: $S_{i,\min} = \operatorname{argmin} \hat{C}_i(S_i)$
- Exclude solutions $\mathbf{S}$ with $S_i \leq S_{i,\min}$ for some i
- Then the remaining problem can be solved with a greedy algorithm

Optimization (cont.)

Algorithm 2.3 *(Greedy algorithm)*

Step 1 $S_{i,\min} := \arg\min \hat{C}_i(S_i)$ for all $i \in I$;

Set $S_i := S_{i,\min}$ for all $i \in I$, and $\mathbf{S} = (S_{1,\min}, \ldots, S_{|I|,\min})$;

$\mathscr{E} := \{\mathbf{S}\}$;

Compute $\hat{C}(\mathbf{S})$ and $W(\mathbf{S})$.

Step 2 $\Gamma_i := \frac{-\Delta W_i(S_i)}{\Delta \hat{C}_i(S_i)}$ for all $i \in I$;

$k := \arg\max\{\Gamma_i : i \in I\}$;

$\mathbf{S} := \mathbf{S} + \mathbf{e}_k$;

$\mathscr{E} := \mathscr{E} \cup \{\mathbf{S}\}$.

Step 3 Compute $\hat{C}(\mathbf{S})$ and $W(\mathbf{S})$;

If 'stop criterium', then stop, else goto Step 2.

5. METRIC model

(cf. Sherbrooke [1968])

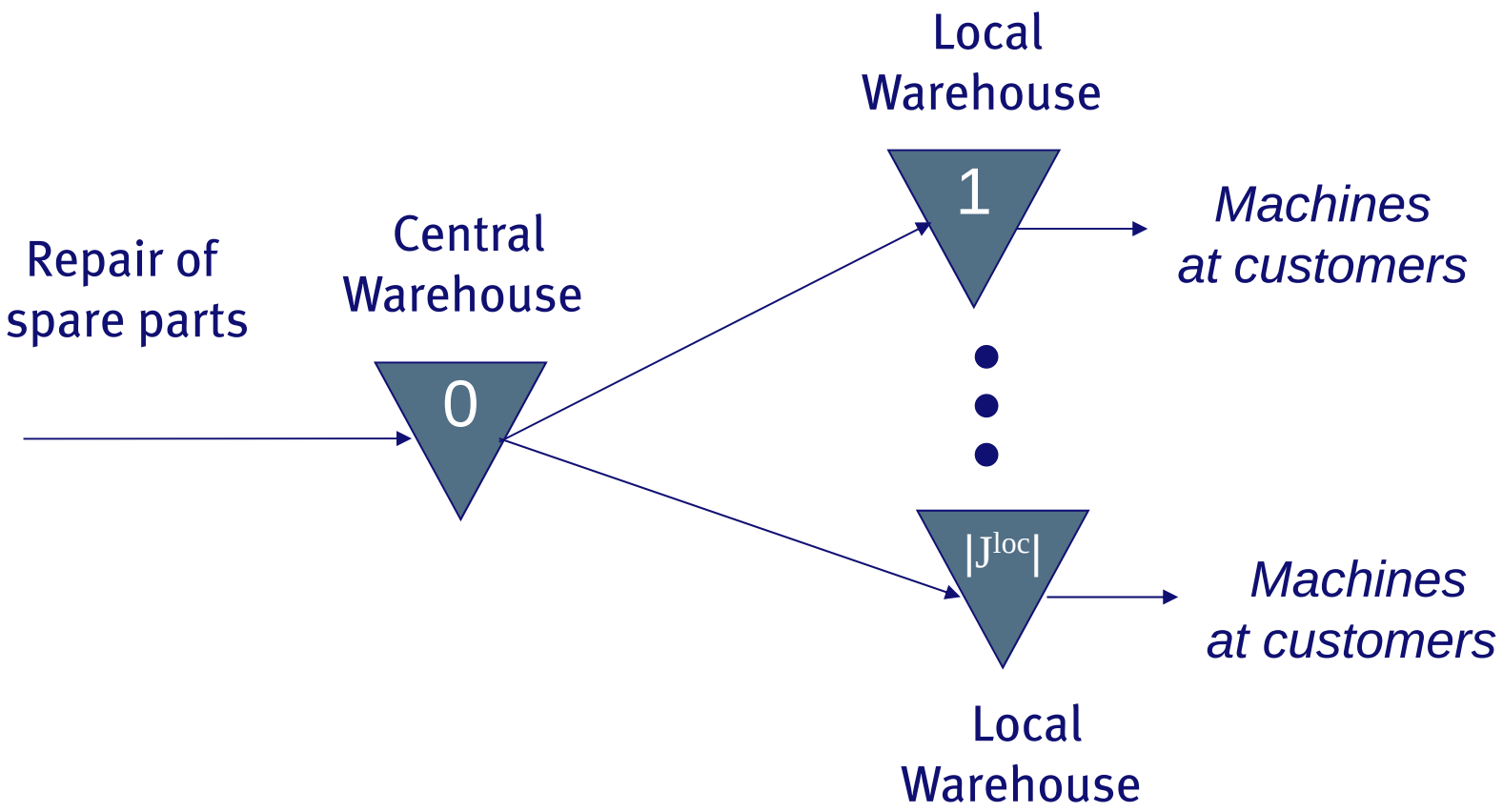
5.1 Model description

5.2 Evaluation

5.3 Optimization

5.1. Model description

Model


Local
Warehouse
1
Machines
at customers
Repair of
spare parts
Central
Warehouse
0
|J^loc|
Machines
at customers
Local
Warehouse

Notation

Input variables:

- J^{loc} : Set of Local Warehouses (LW's), LW's are numbered 1,..., $|J^{\text{loc}}|$
- 0 : Index for Central Warehouse (CW)
- J : Set of all warehouses, $J = \{0\} \cup J^{\text{loc}}$
- I : Set of critical SKU's, SKU's are numbered 1,..., $|I|$
- $m_{i,j}$: Demand rate for SKU i at LW j ($m_{i,j} \geq 0$)

Notation (cont.)

Input variables (cont.):

- $t_{i,j}$: Order and ship time from CW to LW j for SKU i (these times are deterministic; $t_{i,j} > 0$)
- $t_{i,0}$: Mean repair leadtime for SKU i (repair leadtimes are i.i.d.; $t_{i,0} > 0$)
- c_i^{h} : Holding cost rate per part of SKU i ($c_i^{\mathrm{h}} > 0$)
- EBO_j^{obj} : Target level for aggregate mean number of backorders at LW j

Notation (cont.)

Decision variables:

- $S_{i,j}$: Basestock level for SKU i at warehouse j $(S_{i,j} \in \{0,1,\ldots\})$
- $\mathbf{S}_j = (S_{1,j},\ldots, S_{|I|,j})$: Basestock vector for SKU i
- $\mathbf{S}$: Matrix with all basestock levels

$$\mathbf{S} = \begin{pmatrix} S_{1,0} & S_{1,1} & \ldots & S_{1,|J^{loc}|} \\ S_{2,0} & S_{2,1} & \ldots & S_{2,|J^{loc}|} \\ \vdots & \vdots & \ddots & \vdots \\ S_{|I|,0} & S_{|I|,1} & \ldots & S_{|I|,|J^{loc}|} \end{pmatrix}$$

Notation (cont.)

Output and other variables (cont.):

- $EBO_{i,j}(S_{i,0}, S_{i,j})$: Mean number of backorders for SKU i at LW j;
- $EBO_j(\mathbf{S}_0, \mathbf{S}_j)$: Aggregate mean number of backorders at LW j

$$EBO_j(\mathbf{S}_0, \mathbf{S}_j) = \sum_{i \in I} EBO_{i,j}(S_{i,0}, S_{i,j}).$$

Notation (cont.)

Output and other variables (cont.):

- $C(\mathbf{S}) = \Sigma_{i \in I} \Sigma_{j \in J} c_i^{\mathrm{h}} S_i$: Total average costs (excl. holding costs for pipeline stock)

Optimization problem

Problem (R)*:*

$$\min \quad C(\mathbf{S})$$

subject to

$$EBO_j(\mathbf{S}_0, \mathbf{S}_j) \le EBO_j^{\text{obj}}, \quad j \in J^{\text{loc}},$$

$$S_{i,j} \in \{0,1,\ldots\} \quad \text{for all } i \in I \text{ and } j \in J$$

Remark: Sherbrooke [1968] had a constraint for sum of $EBO_j(\mathbf{S}_0, \mathbf{S}_j)$. Then a greedy procedure can be applied to generate efficient solutions (after convexification).

5.2. Evaluation

Exact Evaluation (due to Graves [1985])

Observation 1:

Evaluation can be done per SKU i

(the SKU's are only 'connected' via the aggregate mean waiting times)

Exact evaluation (cont.)

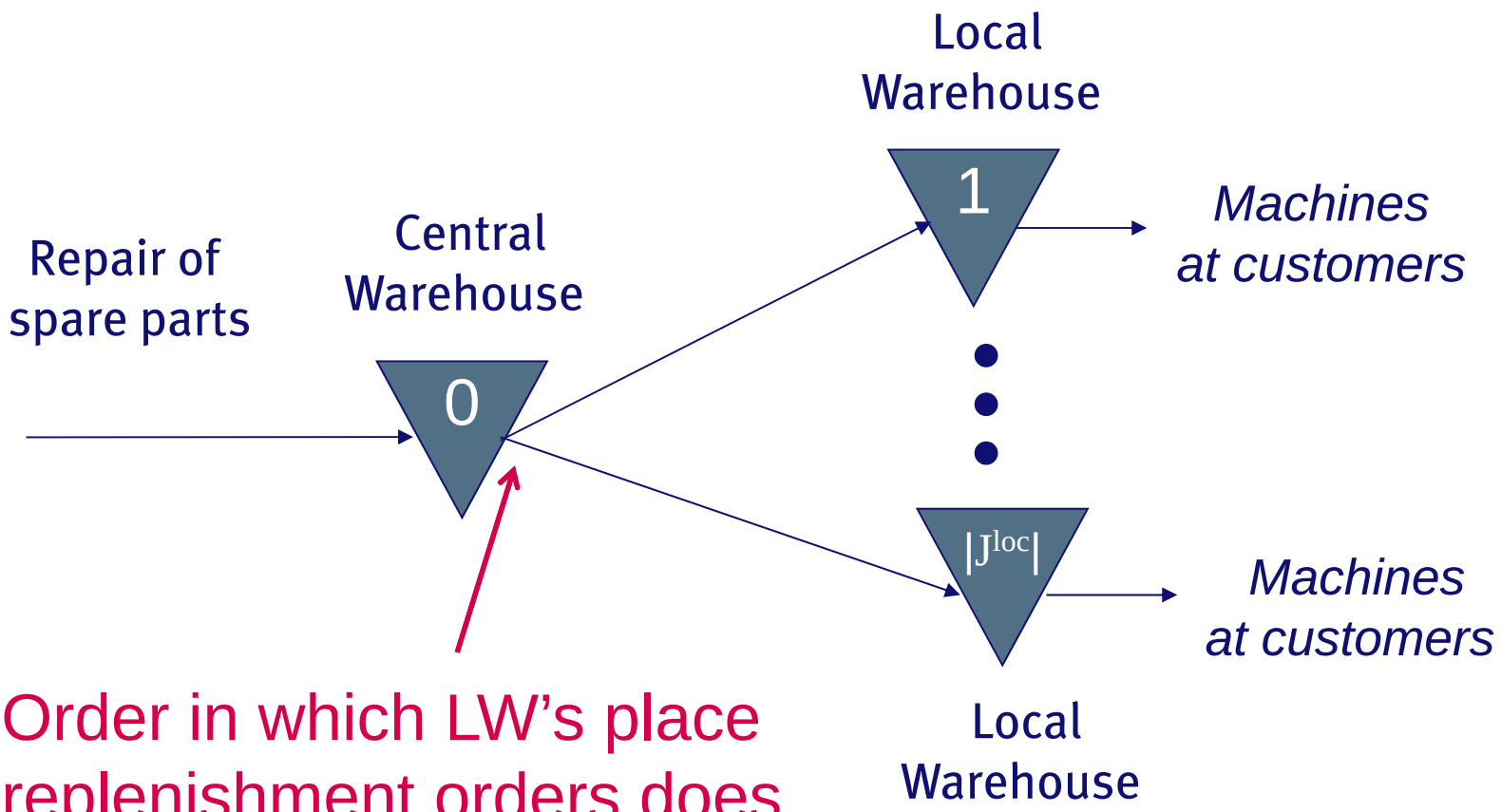
Observation 2:

For each SKU $i \in I$:

- LW places replenishment orders at CW j according to a Poisson process with rate $m_{i,j}$
- The total demand process at CW is a Poisson process with rate:

$$m_{i,0} = \sum_{j \in J^{\text{loc}}} m_{i,j}$$

Exact evaluation (cont.)



Order in which LW's place replenishment orders does not depend on the basestock levels !

Exact evaluation (cont.)

Line of analysis – Variables that are determined:

- $I_{i,0}(S_{i,0})$: On-hand stock of SKU i at CW (in steady state)
- $B_{i,0}(S_{i,0})$: Number of backorders for SKU i at CW (in steady state)
- $B_{i,0}^{(j)}(S_{i,0})$: Number of backorders for SKU i of LW j at CW (in steady state)
- $I_{i,j}(S_{i,0},S_{i,j})$: On-hand stock of SKU i at LW j (in steady state)
- $B_{i,j}(S_{i,0},S_{i,j})$: Number of backorders for SKU i at LW j (in steady state)

Same variables with E added j: **mean** value

Exact Evaluation (cont.)

Given: $i \in I$

Define:

$Y_{i,j}$: Poisson distributed random variable with mean $m_{i,j}\, t_{i,j}$ for each warehouse j

Exact Evaluation (cont.)

Step 1:

- $X_{i,0}$: Number of parts in repair pipeline at CW
- $I_{i,0}(S_{i,0}) = (S_{i,0} - X_{i,0})^+$ and $B_{i,0}(S_{i,0}) = (X_{i,0} - S_{i,0})^+$

- Repair shop at CW is as a M|G|∞ queue, and thus Palm's theorem may be applied
- Thus: $X_{i,0} = Y_{i,0}$

Exact Evaluation (cont.)

Step 2:

- Due to FCFS allocation:

  Each backordered demand at CW is a demand from LW j with probability $m_{i,j}/m_{i,0}$

- Hence:

$$\mathbb{P}\left\{B_{i,0}^{(j)}(S_{i,0}) = x\right\}$$
$$= \sum_{y=x}^{\infty} \binom{y}{x} \left(\frac{m_{i,j}}{m_{i,0}}\right)^{x} \left(1 - \frac{m_{i,j}}{m_{i,0}}\right)^{y-x} \mathbb{P}\left\{B_{i,0}(S_{i,0}) = y\right\}.$$

Exact Evaluation (cont.)

Step 3:

- $X_{i,j}(S_{i,0})$: Number of parts in replenishment pipeline at LW j
- $I_{i,j}(S_{i,0},S_{i,j}) = (S_{i,j} - X_{i,j}(S_{i,0}))^+$ and $B_{i,j}(S_{i,0},S_{i,j}) = (X_{i,j}(S_{i,0}) - S_{i,j})^+$
- Due to deterministic order and ship time:

  $X_{i,j}(S_{i,0})$ at time t = $B_{i,0}^{(j)}(S_{i,0})$ at time $t - t_{i,j}$

  + Demand in interval $[t - t_{i,j}, t)$
- Hence: $X_{i,j}(S_{i,0}) = B_{i,0}^{(j)}(S_{i,0}) + Y_{i,j}$

Appr. eval. via two-moment fits (cf. Graves [1985])

Procedure:

- Compute first two moments of $I_{i,0}(S_{i,0})$ and $B_{i,0}(S_{i,0})$
- Compute first two moments of $B_{i,0}^{(j)}(S_{i,0})$ for each LW j:

  simple formulas available
- Compute first two moments of $X_{i,j}(S_{i,0})$ for each LW j
- Fit a negative Binomial distribution on first two moments of $X_{i,j}(S_{i,0})$ for each LW j
- Compute first moments of $I_{i,j}(S_{i,0},S_{i,j})$ and $B_{i,j}(S_{i,0},S_{i,j})$

=> Accurate and efficient approximation!

METRIC approach (cf. Sherbrooke [1985])

Procedure:

- Compute first moments of $I_{i,0}(S_{i,0})$ and $B_{i,0}(S_{i,0})$
- Compute first moment of $B_{i,0}^{(j)}(S_{i,0})$ for each LW j:

  mean of $B_{i,0}^{(j)}(S_{i,0})$ = $(m_{i,j} / m_{i,0}) \times$ (mean of $B_{i,0}(S_{i,0})$)
- Compute first moment of $X_{i,j}(S_{i,0})$ for each LW j
- Fit a Poisson distribution on first moment of $X_{i,j}(S_{i,0})$ for each LW j
- Compute first moments of $I_{i,j}(S_{i,0},S_{i,j})$ and $B_{i,j}(S_{i,0},S_{i,j})$

=> Not always accurate!

5.3. Optimization

Greedy heuristic (cf. Wong et al. [2007])

Main idea:

- Start with zero stock for all SKU's at all LW's
- Use a steepest descent method to decrease costs
- Use a greedy logic to get to a feasible solution

Greedy heuristic (cont.)

- Distance of a solution $\mathbf{S}$ to the set of feasible solutions:

$$\sum_{j \in J^{\mathrm{loc}}} \left(EBO_j(\mathbf{S}) - EBO_j^{\mathrm{obj}} \right)^+ .$$

Greedy heuristic (cont.)

- Decrease in distance to feasible solutions per unit of increase in costs:

$$\Gamma_{i,j} = \Delta_{i,j}EBO / c_i^h.$$

$$\Delta_{i,j}EBO = \sum_{l \in J^{loc}} \left[\left(EBO_l(\mathbf{S}) - EBO_l^{obj} \right)^+ - \left(EBO_l(\mathbf{S} + e_{i,j}) - EBO_l^{obj} \right)^+ \right],$$

Greedy heuristic (cont.)

Algorithm 4.1 (*Greedy Algorithm*).

Step 1. $S_{i,j} := 0$ for all $i \in I, j \in J$ (so $\mathbf{S}_j := (0, \ldots, 0)$ for all $j \in J$ and $\mathbf{S} := (\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{|J^{\text{loc}}|})$);

$C(\mathbf{S}) := 0$ and $EBO_j(\mathbf{S}) := \sum_{i \in I} m_{i,j}(t_{i,0} + t_{i,j})$ for all $j \in J^{\text{loc}}$.

Step 2. $\Gamma_{i,j} := \frac{\Delta_{i,j} EBO}{c_i^{\text{h}}}$ for all $i \in I, j \in J$;

$(k, l) := \arg\max_{(i,j) \in I \times J} \Gamma_{i,j}$;

$\mathbf{S} := \mathbf{S} + e_{k,l}$.

Step 3. $C(\mathbf{S}) := C(\mathbf{S}) + c_k^{\text{h}}$;

Calculate $EBO_j(\mathbf{S})$ for all $j \in J^{\text{loc}}$;

If $EBO_j(\mathbf{S}) \leq EBO_j^{\text{obj}}$ for all $j \in J^{\text{loc}}$, then stop, else go to step 2.

Leads to: Heuristic solution (under exact evaluations)

No theoretical results like for single-location model

Greedy heuristic (cont.)

Dantzig-Wolfe decompostion leads to:

- Lower bound
- Heuristic solution

} Under exact evaluations

<u>Performance of greedy heuristic:</u>

- Small optimality gap
- # SKU's ↑ => Optimality gap ↓
- Computation time is relatively low

6. Multi-location model with lateral and emergency shipments

General setting

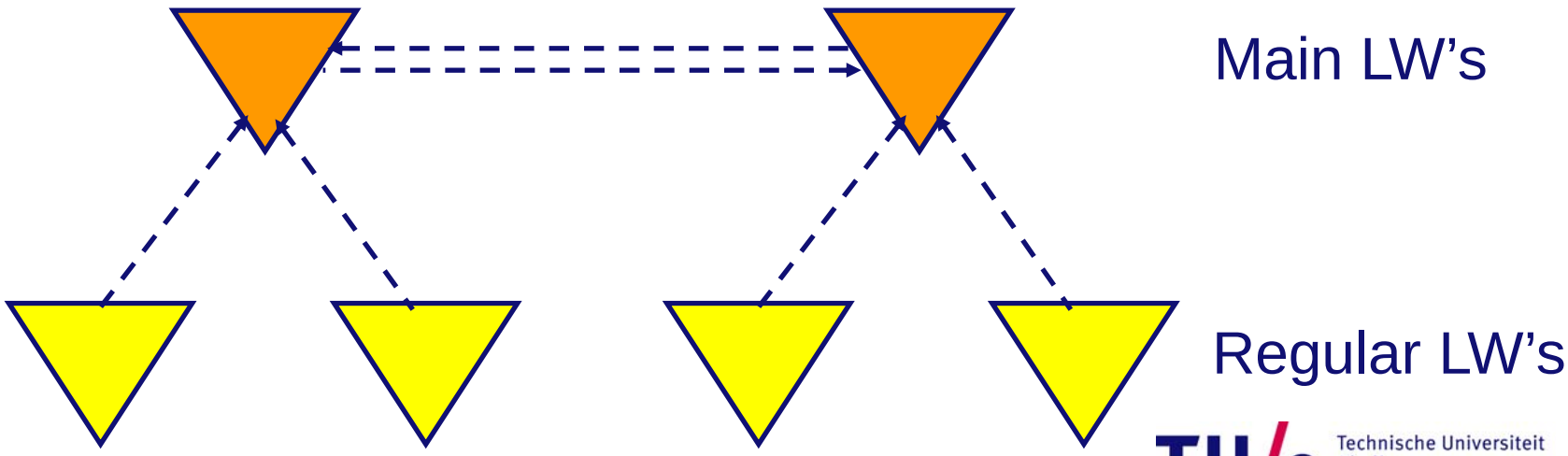
Key features:

- Multiple local warehouses (LW's)
- Each LW supports one or more groups of machines
- Aggregate mean waiting time target per group
- Central warehouse: outside scope of model, is assumed to have infinite stock
- Multiple SKU's
- All SKU's are critical
- In case of stockout: application of lateral transshipment or emergency shipment
- Two types of LW's: Main and regular LW's

General setting (cont.)

Two types of LW's: main and regular LW's

- Only main locals are suppliers for lateral transshipment
- This distinction is made to facilitate implementation in practice
- Pooling: full, partial, none



To be determined

Output variables: *(basic output)*

Demand stream for SKU i at LW j :

- $\beta_{i,j}(\mathbf{S}_i)$: Fraction satisfied by LW j itself
- $\alpha_{i,j,k}(\mathbf{S}_i)$: Fraction satisfied by main LW k via a lateral transshipment
  - $A_{i,j}(\mathbf{S}_i)$: Total fraction satisfied by a lat. transshipm., $A_{i,j}(\mathbf{S}_i) = \Sigma_{k \in K \setminus \{j\}} \alpha_{i,j,k}(\mathbf{S}_i)$
- $\theta_{i,j}(\mathbf{S}_i)$: Fraction satisfied by an emerg. shipment

It holds that: $\beta_{i,j}(\mathbf{S}_i) + A_{i,j}(\mathbf{S}_i) + \theta_{i,j}(\mathbf{S}_i) = 1$

Exact evaluation

Done per given SKU $i \in I$

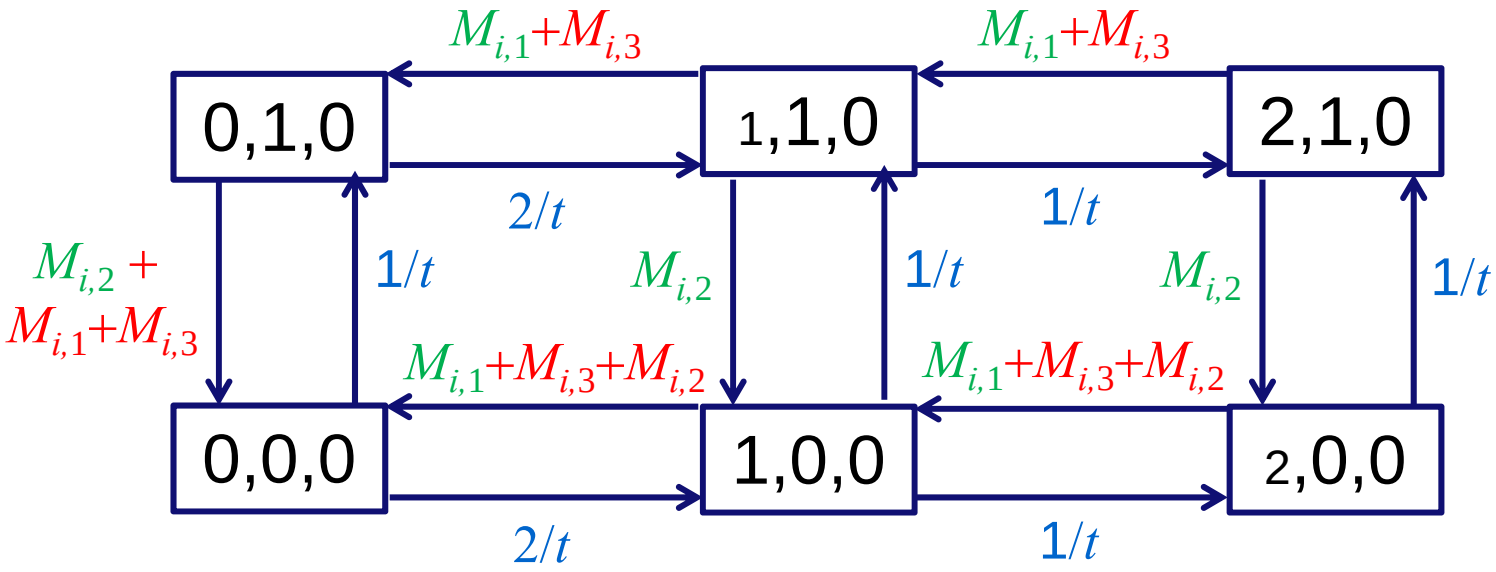
Via Markov process:

- States $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,|J|})$, where $x_{i,j}$ denotes the on-hand stock at LW i
- Steady-state distribution $\{\pi(\mathbf{x}_i)\}$ is determined numerically
- The $\beta_{i,j}(\mathbf{S}_i)$, $\alpha_{i,j,k}(\mathbf{S}_i)$, $\theta_{i,j}(\mathbf{S}_i)$ follow from the $\pi(\mathbf{x}_i)$
- Number of states: $\Pi_{j \in J}\,(S_i + 1)$

EXAMPLE

Given: $J = \{1,2,3\}$, $K = \{1,2\}$, $k_3 = 1$,

$$S_{i,1} = 2,\ S_{i,2} = 1,\ S_{i,3} = 0$$

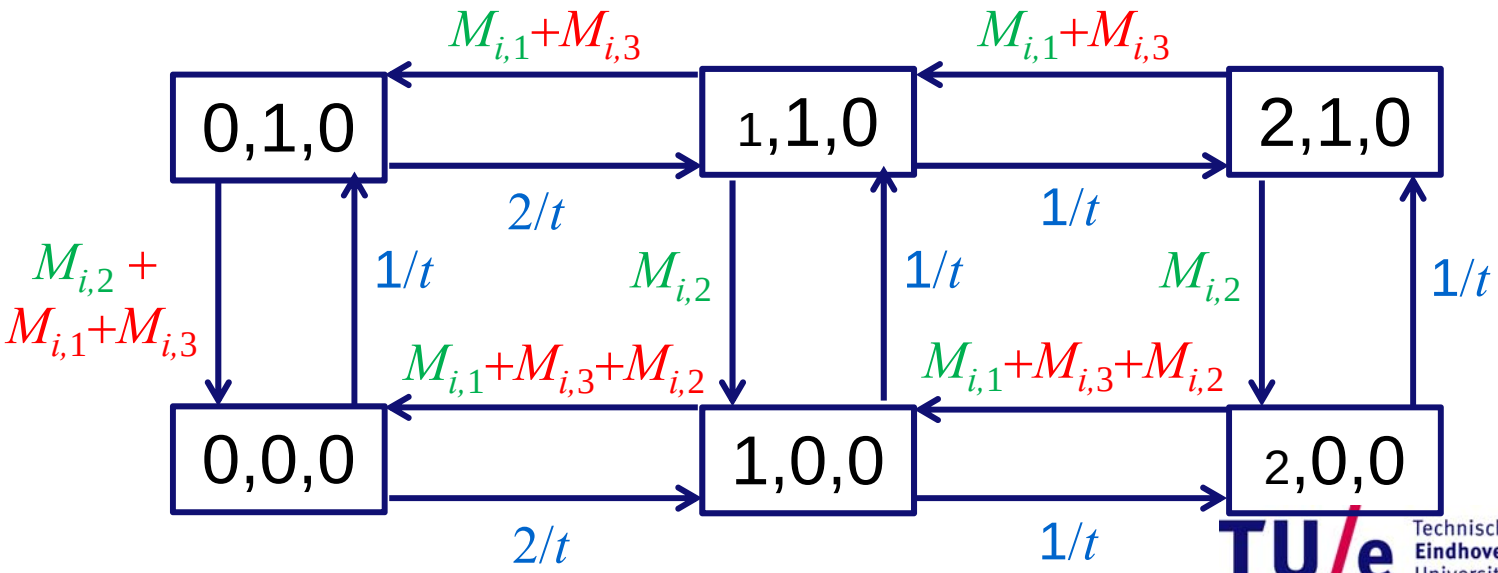


EXAMPLE (cont.)

We obtain:

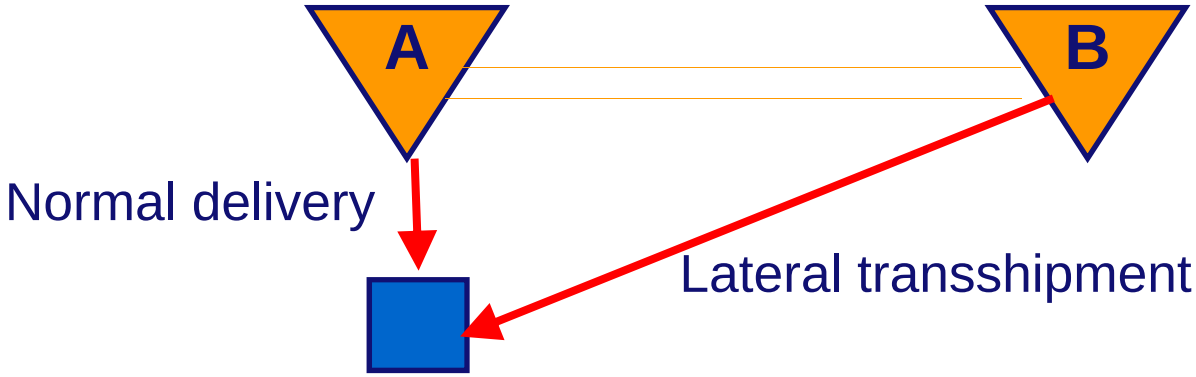
$$\beta_{i,1}(\mathbf{S}_i) = \pi(1,0,0) + \pi(1,1,0) + \pi(2,0,0) + \pi(2,1,0),$$

$$\alpha_{i,1,2}(\mathbf{S}_i) = \pi(0,1,0),\ \theta_{i,1}(\mathbf{S}_i) = \pi(0,0,0),\ \ldots$$



Approximate evaluation: Principle
(cf. Kranenburg and Van Houtum, 2009)

Principle: Decompose network into single-location models



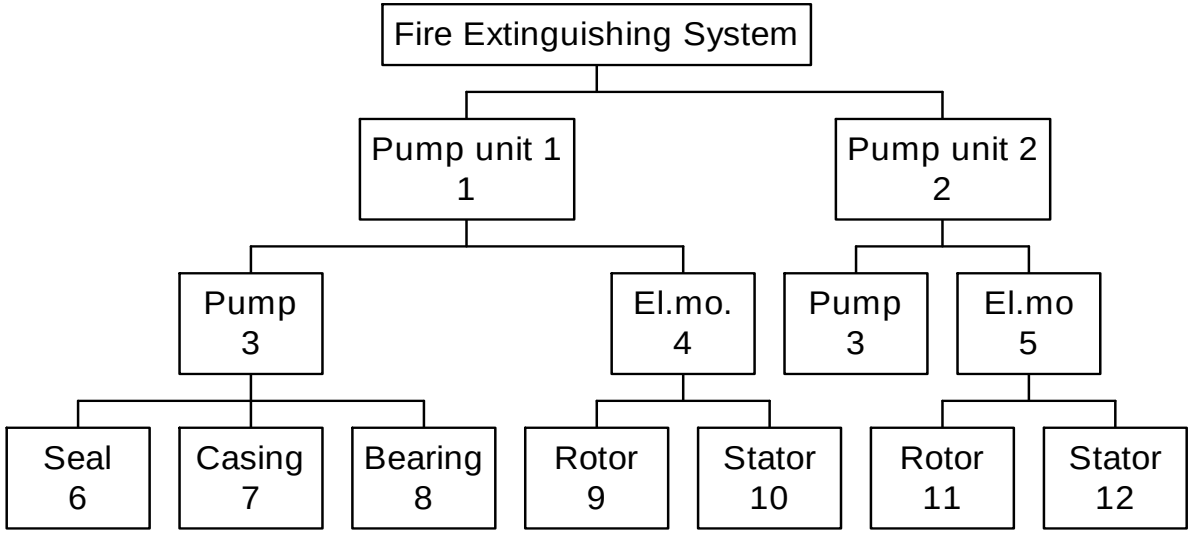
- Normal delivery by A
- If A out of stock: lateral transshipment from B
- Then B observes *extra* demand: overflow demand
- *Approximation: Overflow demand processes are Poisson processes*
- Iterative procedure

Approximate evaluation

- Efficient and accurate
- One can apply the procedure to many structures for the lateral transsipments!
- In use at ASML since 2005
  - 60 stockpoints, 5000 SKU's, say
  - Optimization: Greedy heuristic

7. Extensions

Multi-echelon, multi-indenture systems



- Exact recursion like for METRIC model
- Efficient and accurate appr. Eval. Based on two-moment fits

Batching

Easy to incorporate for:

- Single-location model with backordering
- METRIC model with (s,Q) rule at central warehouse and given Q's

Hard to incorporate for models with emergency shipments

Multiple demand classes

Standard way: Critical levels

Drawbacks:

- More complicated optimization problem; no greedy heuristic available
- Critical levels are too strict for very low demand rates
- Critical levels may not be accepted in practice

Challenge: Find better ways for the differentiation when demand rates are low

8. Applications in practice

Advanced software packages:

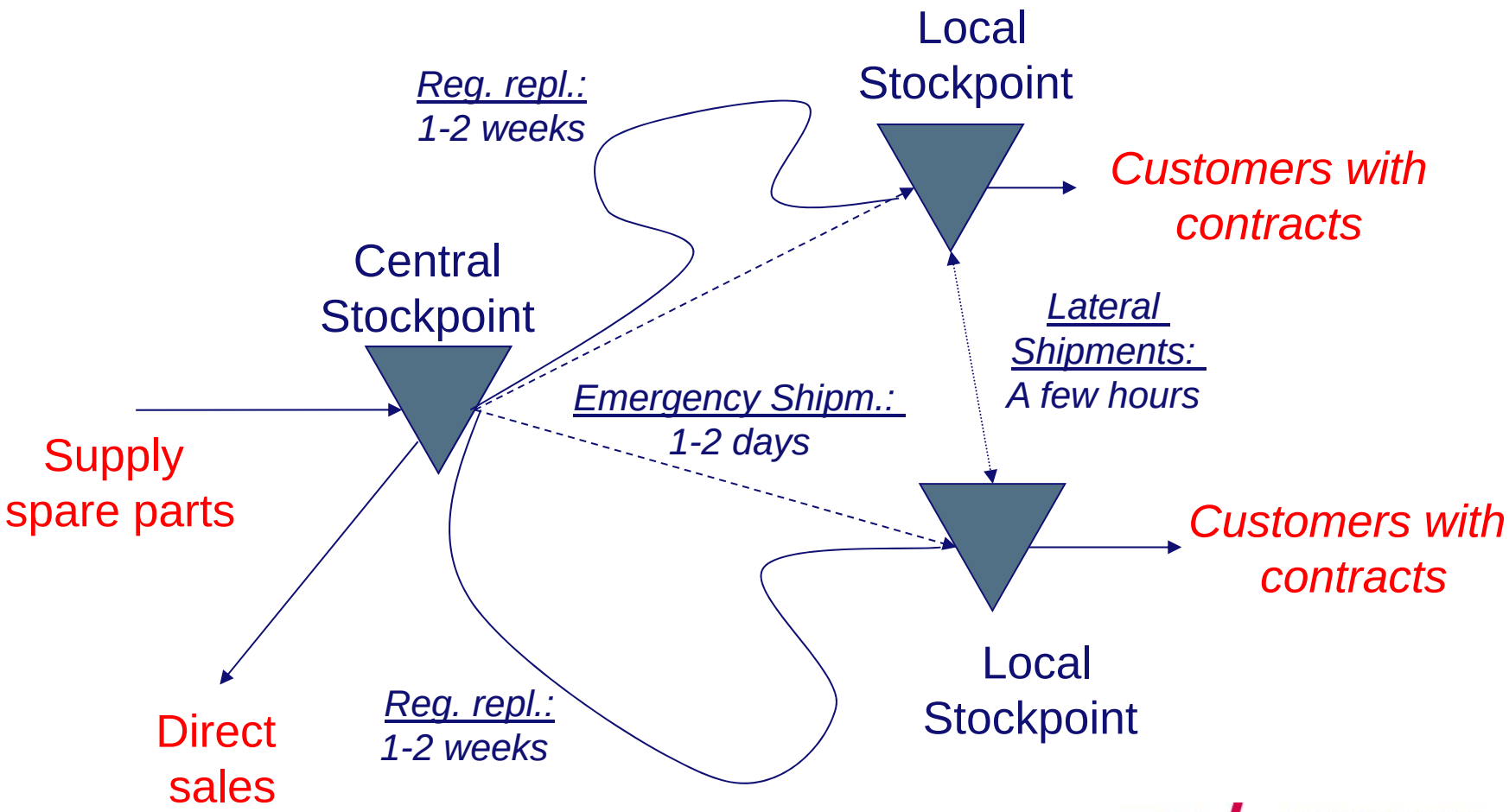
- MCA Solutions: Founded by Morris Cohen & Associates
- Xelus
- Network Neighborhood: Developed by IBM
- …

Own developments:

- US Coast Guard: Development with Deshpande et al. [2006]
- ASML: Planning algorithm developed with TU/e
- …

9. Challenges

The full problem



Dynamic decisions

Motivation:

- One has visibility of actual stocks
- Condition monitoring data
- Service contracts are for 1/2/3 years

Needed:

- Approximate dynamic programming, …
- Interaction between tactical and operational planning

Example of condition monitoring data

MACHINE NUMBER	TIMESTAMP	MACHINE TYPE	CUSTOMER ID	P955	P956	P957	P958	P959	P960	P961
M0005	07-Jan-09	T0007	C1058	-6.8961	-6.8860	-6.8910	-6.9177	-6.8542	-6.8860	-3.7979
M0005	13-Feb-09	T0007	C1058	-7.3892	-7.3831	-7.3862	-7.4487	-7.3123	-7.3805	-4.3121
M0005	16-Feb-09	T0007	C1058	-7.4847	-7.4738	-7.4792	-7.5021	-7.4451	-7.4736	-4.3567
M0005	17-Feb-09	T0007	C1058	-7.5400	-7.5320	-7.5360	-7.5974	-7.4640	-7.5307	-4.3823
M0005	19-Feb-09	T0007	C1058	-7.5305	-7.5201	-7.5253	-7.5479	-7.4915	-7.5197	-4.3793
M0005	01-Mar-09	T0007	C1058	-7.6871	-7.6767	-7.6819	-7.7032	-7.6489	-7.6761	-4.5276
M0005	07-Mar-09	T0007	C1058	-7.7783	-7.7686	-7.7734	-7.7954	-7.7414	-7.7684	-4.6072
M0005	10-Mar-09	T0007	C1058	-7.7222	-7.7109	-7.7165	-7.7396	-7.6819	-7.7107	-4.6219
M0005	06-Apr-09	T0007	C1058	-8.0890	-8.0804	-8.0847	-8.1049	-8.0527	-8.0788	-4.9698
M0006	06-Apr-09	T0007	C1058	-7.9700	-7.9602	-7.9651	-7.9858	-7.9337	-7.9597	-4.9725
M0006	22-Apr-09	T0007	C1058	-8.1674	-8.1635	-8.1654	-8.2214	-8.0994	-8.1604	-5.1675
M0006	01-Jun-09	T0007	C1058	-8.5568	-8.5504	-8.5536	-8.5730	-8.5239	-8.5484	-5.6113
M0006	09-Jun-09	T0007	C1058	-8.5599	-8.5553	-8.5576	-8.6090	-8.4949	-8.5519	-5.6860
M0006	22-Jul-09	T0007	C1058	-8.7384	-8.7350	-8.7367	-8.7869	-8.6755	-8.7312	-6.2033
M0006	24-Jul-09	T0007	C1058	-8.8330	-8.8257	-8.8293	-8.8461	-8.8004	-8.8232	-6.1697
M0006	14-Aug-09	T0007	C1058	-8.9163	-8.9142	-8.9153	-8.9639	-8.8553	-8.9096	-6.3815
M0006	14-Aug-09	T0007	C1058	-8.9392	-8.9371	-8.9381	-8.9871	-8.8776	-8.9323	-6.3531
M0006	16-Aug-09	T0007	C1058	-8.8599	-8.8574	-8.8586	-8.9078	-8.7988	-8.8533	-6.4169

Further challenges

- Coupling with service tools planning
- Other performance measures: Long delays vs. Short delays
- Setting of leadtimes for procurement/repair
- Choice of transport modes
- ...