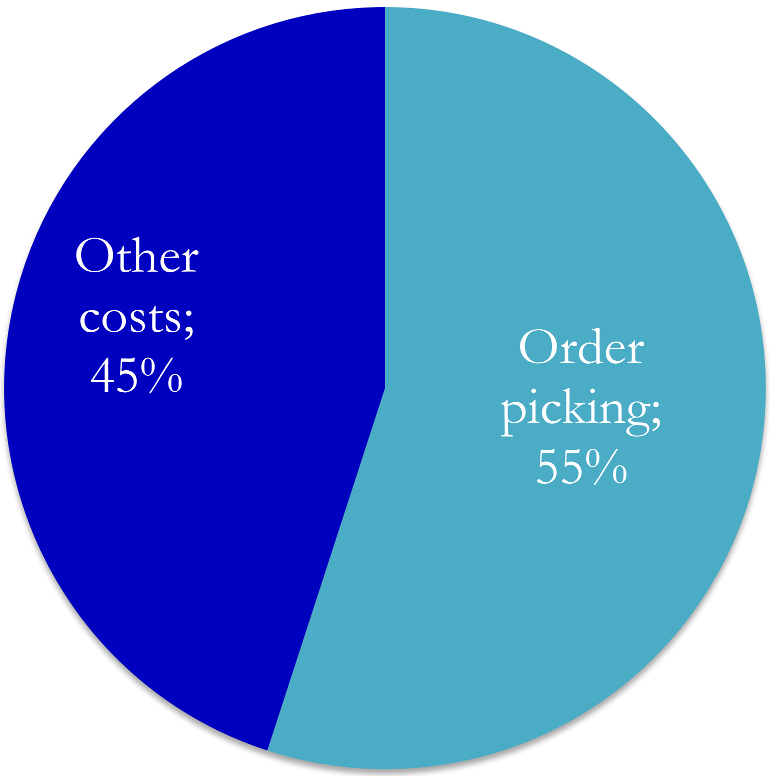
ROTTERDAM SCHOOL OF MANAGEMENT
ERASMUS UNIVERSITY

How to design the most efficient pick and pass system

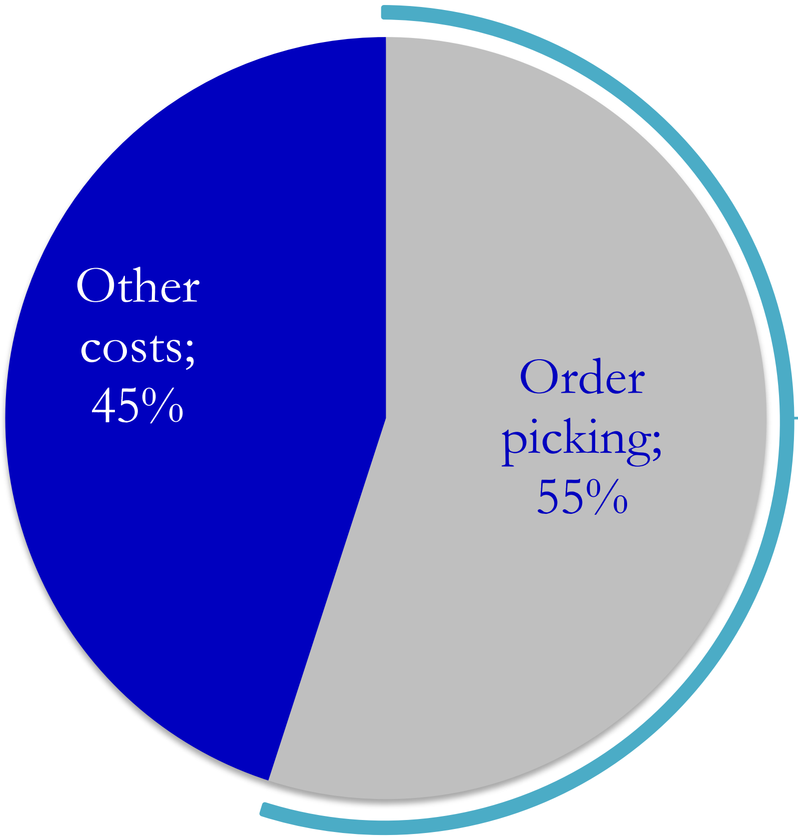
Alina Stroie
MSc Supply Chain Management
Rotterdam School of Management

Warehouse operating costs



Source: Tompkins et al.,2003

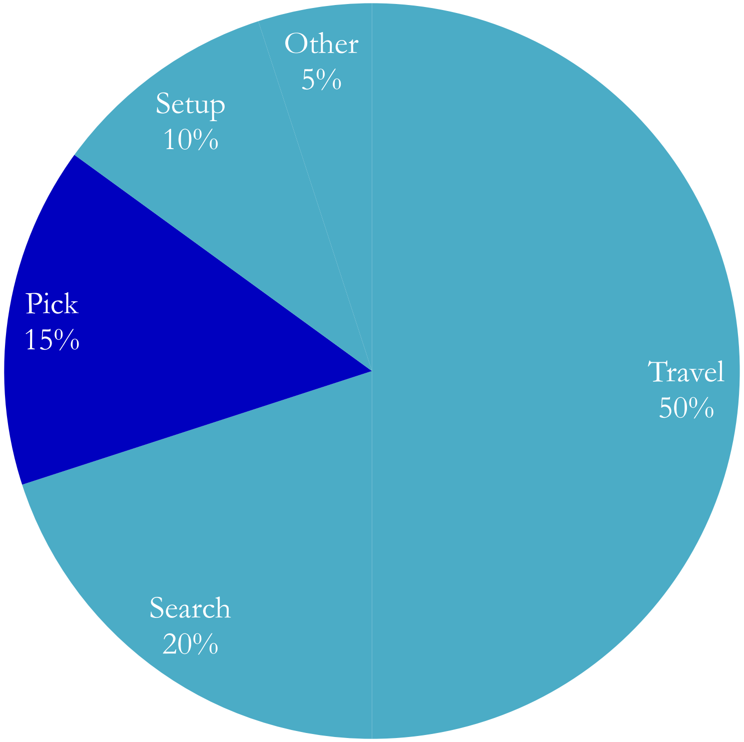
Warehouse operating costs



Significant impact on bottom line
Labor intensive

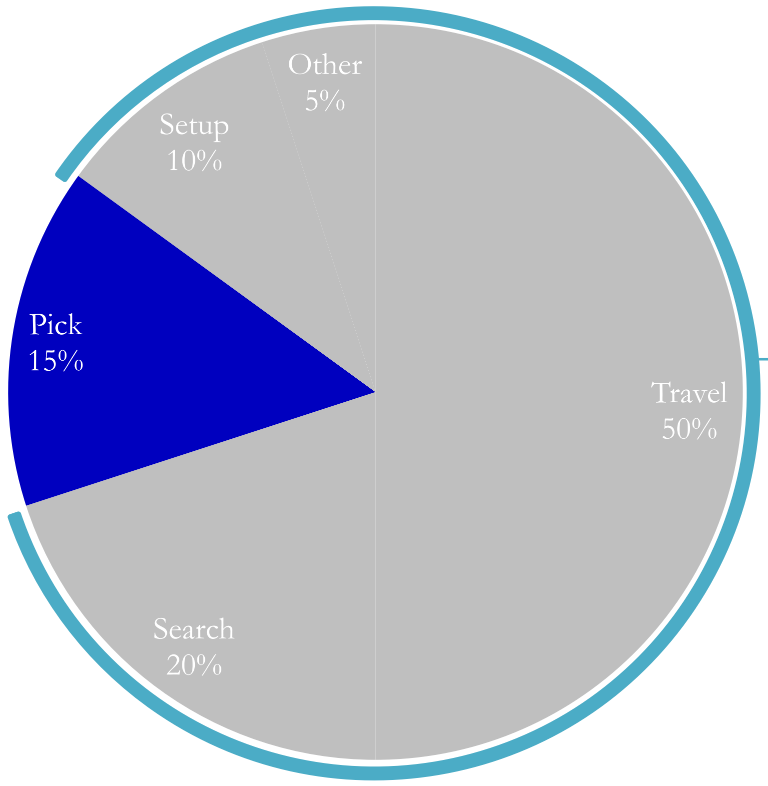
Source: Tompkins et al.,2003

Breakdown of order picker time



Source: Tompkins et al.,2003

Breakdown of order picker time



How can we reduce non-value added time?

Source: Tompkins et al.,2003

Multiple strategies

Pick-and-pass

Simultaneous picking

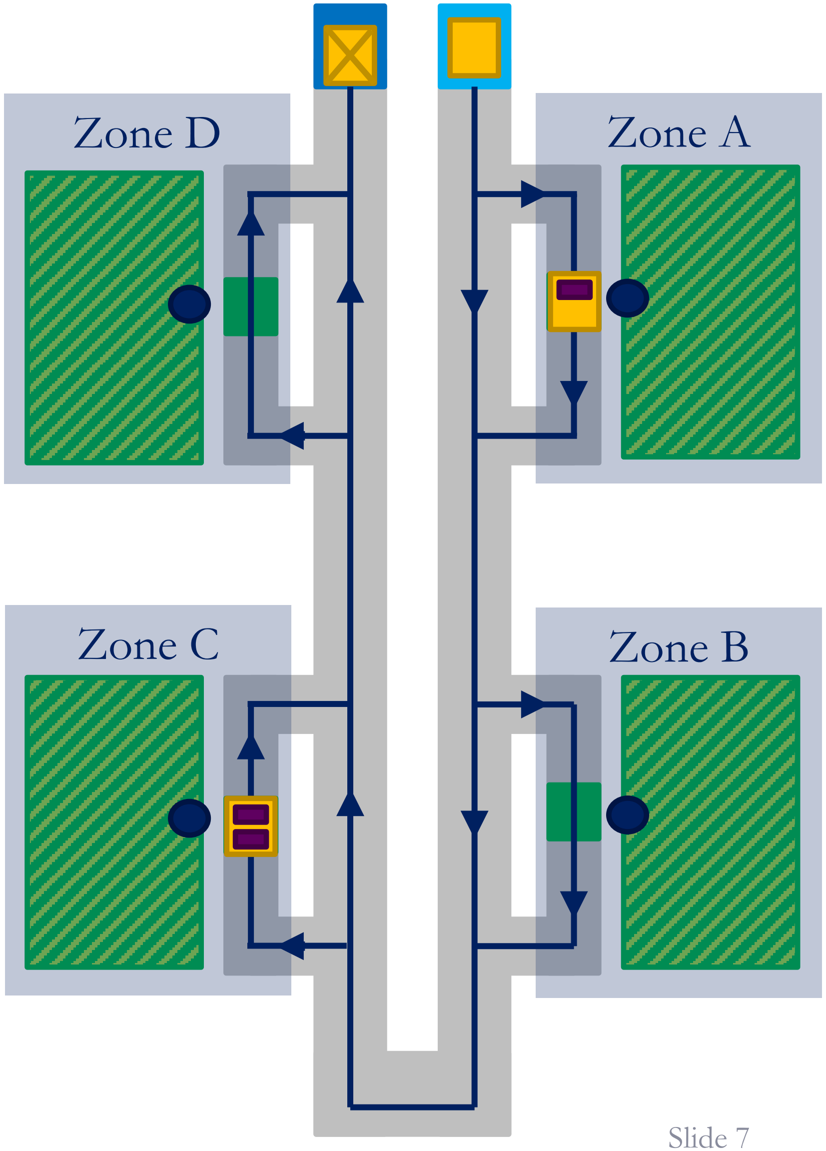
Bucket brigades

- Popular due to simplicity
- Relatively low implementation cost
- High demand rate
- High SKU variety
- Small-medium products

Pick and Pass Zone Picking

- To reduce order picking costs, the storage area is divided into **zones**, covered by one or more pickers; this reduces travel and search time.

- **Pick-and-pass** is an order picking strategy:
  - Each customer order is assigned a tote;
  - Totes visit zones in the system;
  - At each zone, a picker will retrieve products from storage and fill the tote before sending the tote back on the conveyor.



What is the most efficient pick and pass system?

Designing a pick and pass system

Strategic

Tactical

Operational

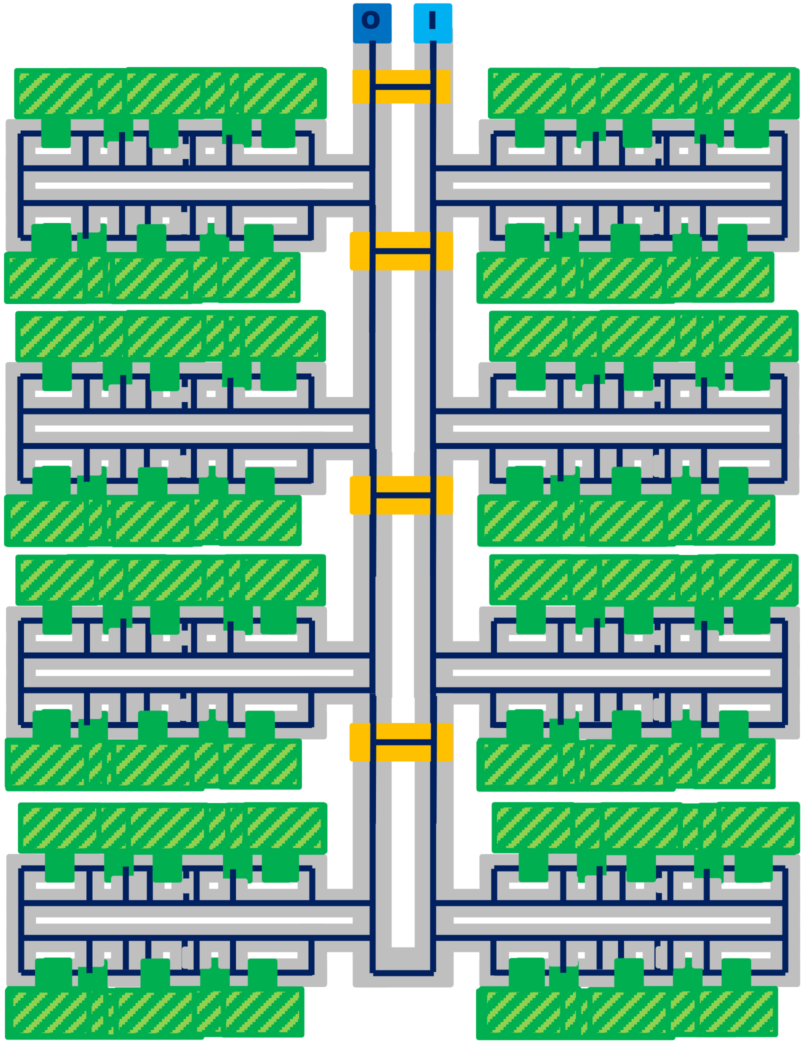
- Number of segments
- Number of zones / segment
- Use of shortcuts
- Block and recirculate protocol
- Class-based versus random storage

What is the most efficient pick and pass system in terms of these 5 variables?

Modelling pick and pass systems

Number of segments	4, 6, 8
Number of zones/segments	2, 4, 6
Allowing totes to recirculate	Yes or No
Allowing totes to use shortcuts	Yes or No
Storage policy	Random or Class-based storage

Modelled in **Material Handling Simulation Package (MHSP)**



Simulation Demo

Material Handling Simulation Program

Simulation parameters

User determined parameters

Order size	UNIF (1,5)
Order generation rate	EXP (0.0167) totes/second
Number of orders per simulation run	1,000
Number of simulations per policy set	20
Picking time	EXP (0.2) picks/second

Reasoning and constraints

- The number of orders per simulation run and the number of simulations per design are selected based on the standard deviation of the average throughput rate. At this setup, the confidence intervals are sufficiently small.
- The order generation rate cannot be higher, otherwise a blockage occurs in specific designs.

Finding the most efficient design

Choice to use Data Envelopment Analysis (DEA)

Minimum set of assumptions to evaluate models along different measurement units (in this case time and cost).

The goal is to compare policy sets with one another, rather than to find the optimal design for a pick-and-pass system; DEA achieves this by ranking policy sets according to their relative efficiency within the group.

Finding the most efficient design

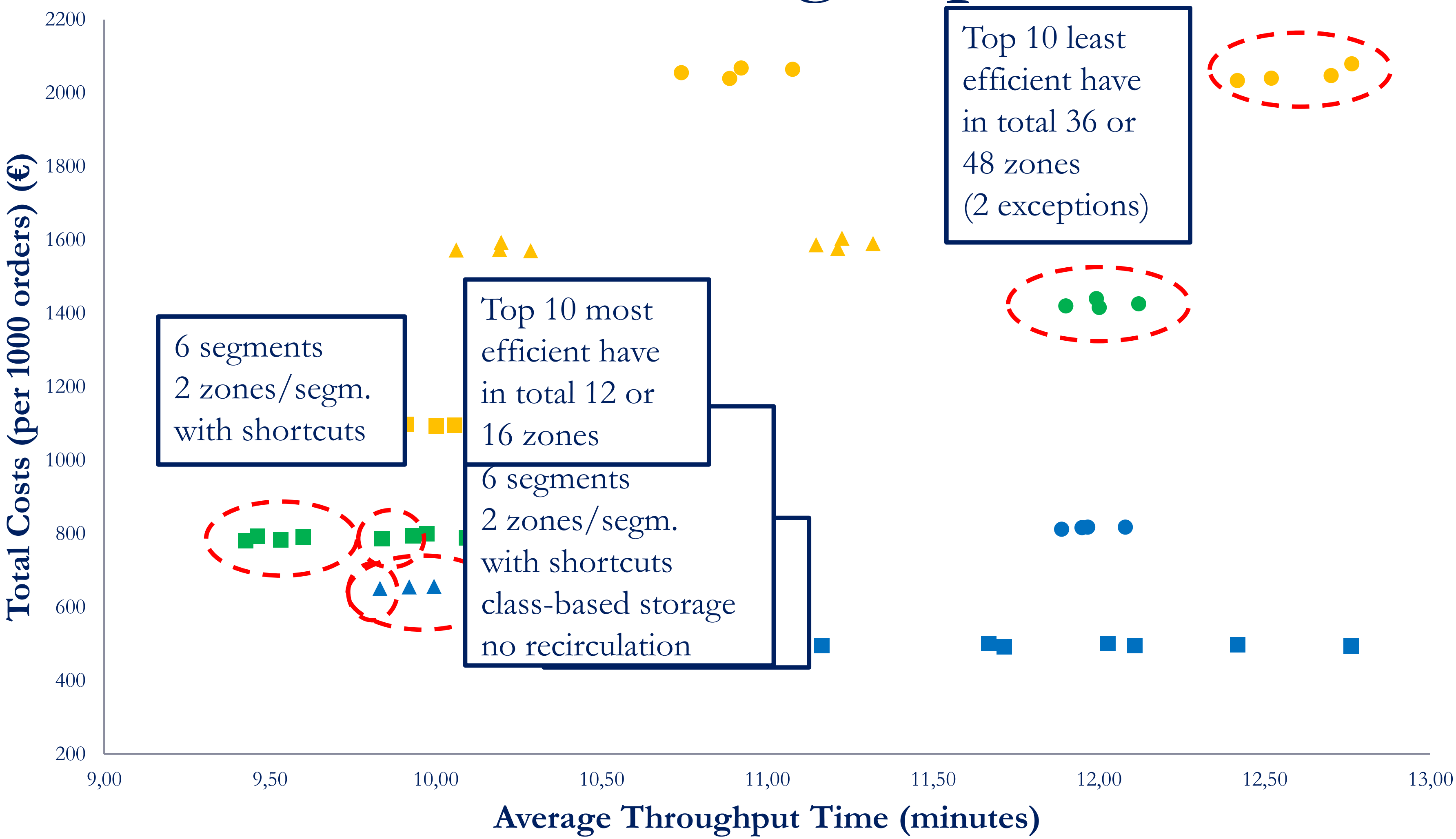
How does DEA work?

DEA calculates the efficiency of a particular policy set based on its ability to generate output given a specific input.

In this case, DEA assigns an efficiency score based on the ability to achieve the lowest average throughput time with the lowest total costs possible.

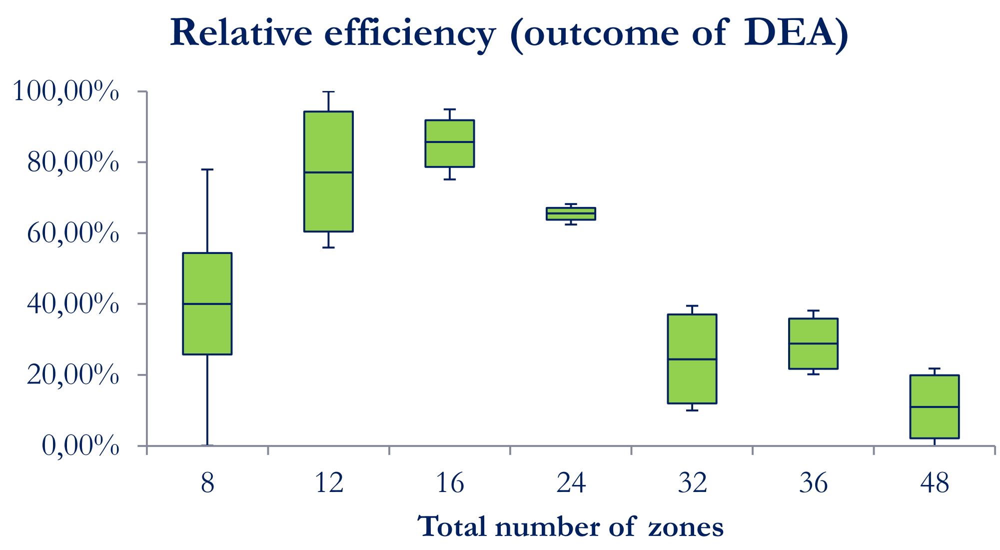
- DEA input is the total cost of a policy set
- DEA output is based on the average throughput time of a policy set

How did the designs perform?

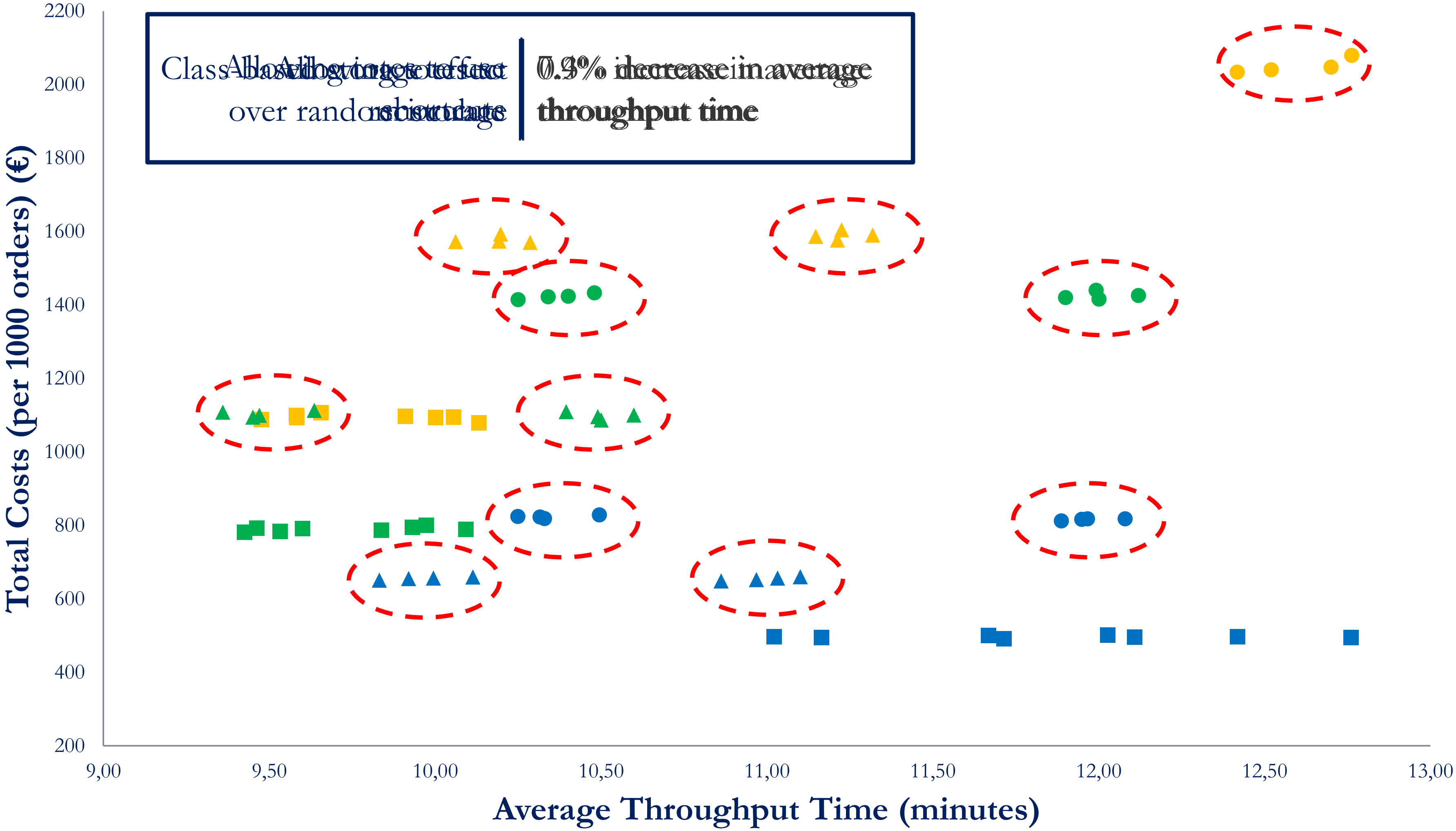


What did I find?

- Designs with **few zones and shortcuts are most efficient** in the set
- Across all metrics studied (total, investment, and operational cost, average throughput time, and make span), a smaller number of zones seems to perform better than designs with many zones.



Some interesting clusters…



When designing your system…

- **No trade-off between total cost and average throughput time** performance: fewer zones perform better on both metrics.
- **Number of zones needs to be able to handle demand**, otherwise the system becomes unstable and performs poorly.
- If possible, **shortcuts** should be implemented as these significantly shorten the time travelled by totes.
- **Storage policy and allowing recirculation** should be decided on a case-by-case basis.

Thank you

Appendix

Calculation of average throughput time

$$Average\ throughput\ time\ simulation_j = \frac{\sum_{i=1}^{1000} Throughput\ time\ tote_i}{1000}$$

$$Average\ policy\ set\ throughput\ time = \frac{\sum_{j=1}^{72} Average\ throughput\ time\ simulation_j}{72}$$

DEA output

To reverse the goal of maximization of the output, for each DMU, the average throughput time of that DMU is subtracted from the maximum average throughput time of all policy sets:

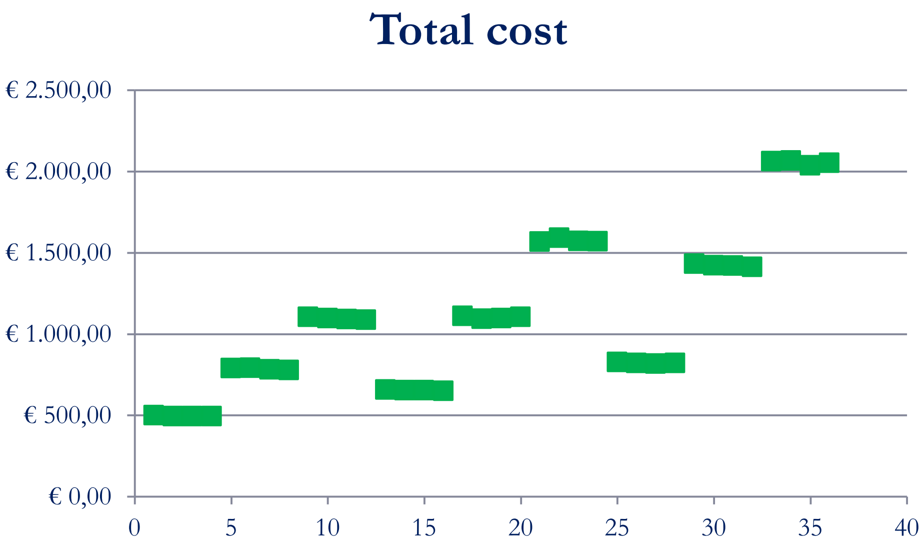
Let avg_i denote the average throughput time of DMU_i.

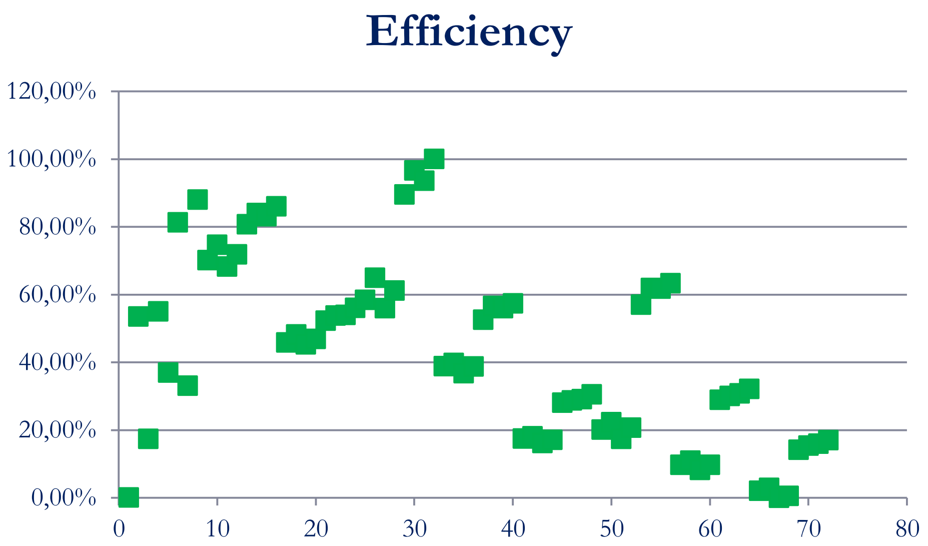
Let $\max = \max(avg_1, avg_{2,} \dots, avg_{72)}$

Then for each DMU the output variable is given by:

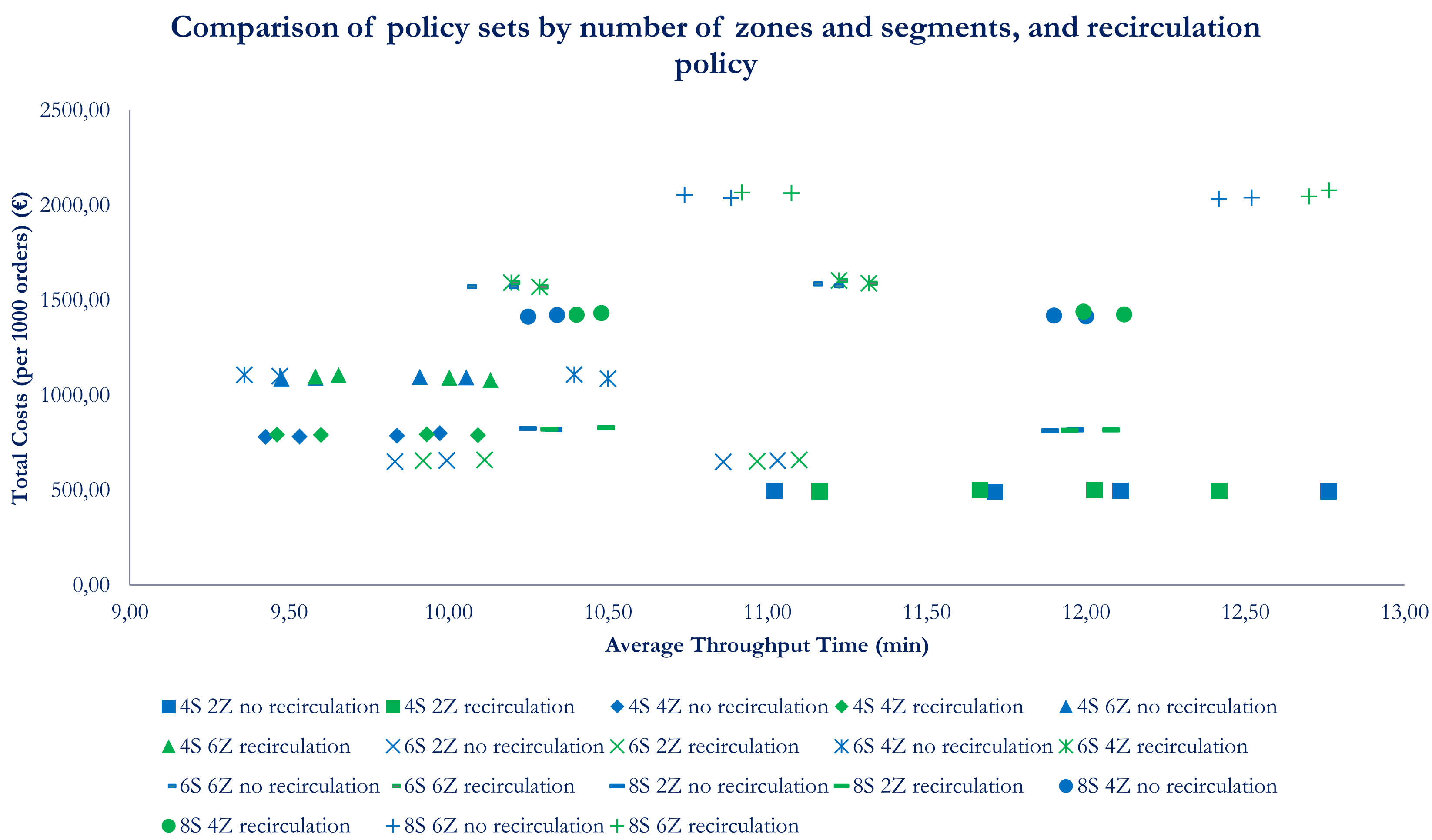
$$(\max - avg_i)$$

Effect of average throughput time on the efficiency of the models

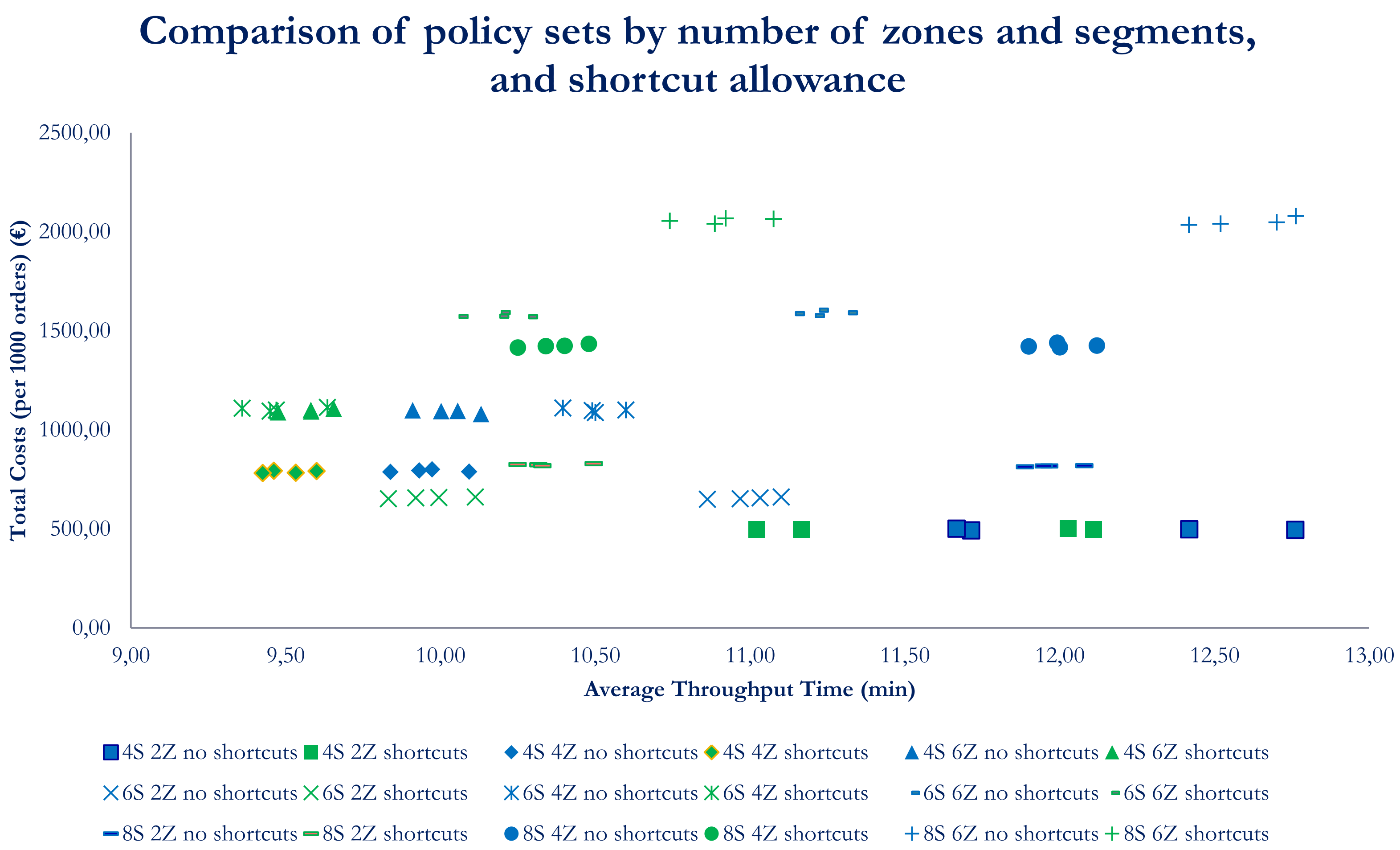




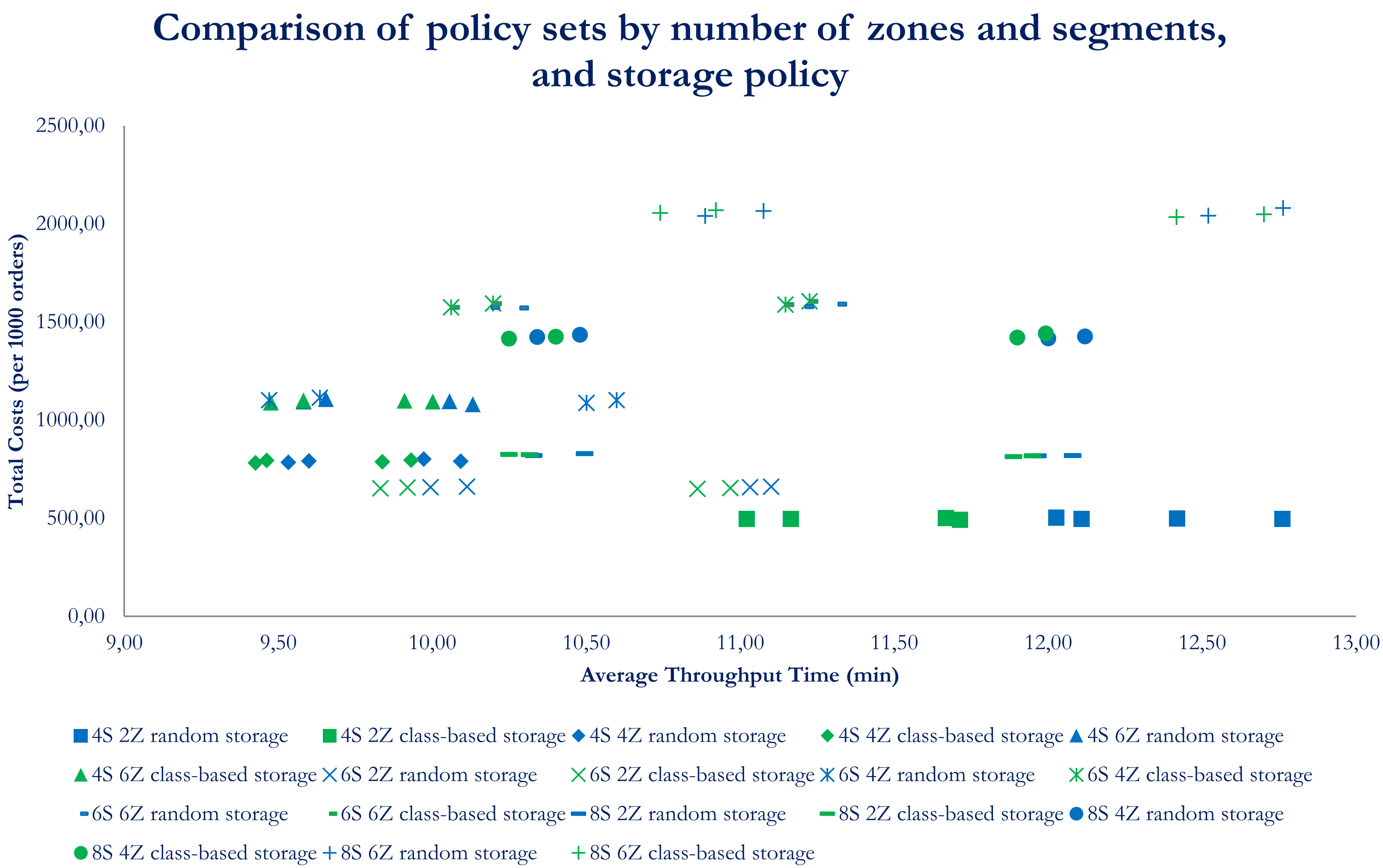
Results: recirculation



Results: shortcuts

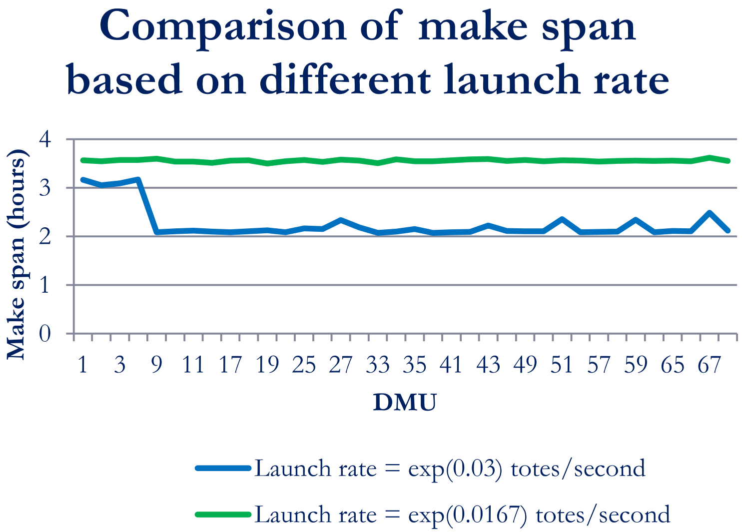


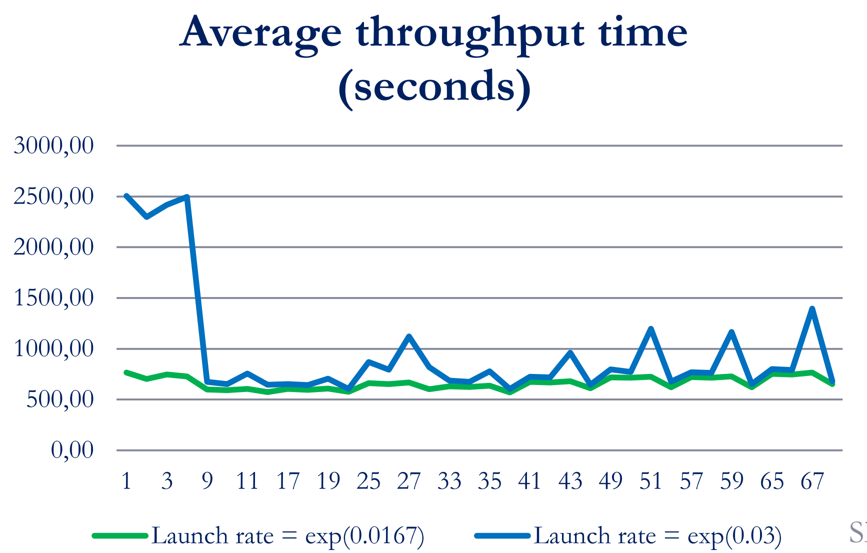
Results: storage policy

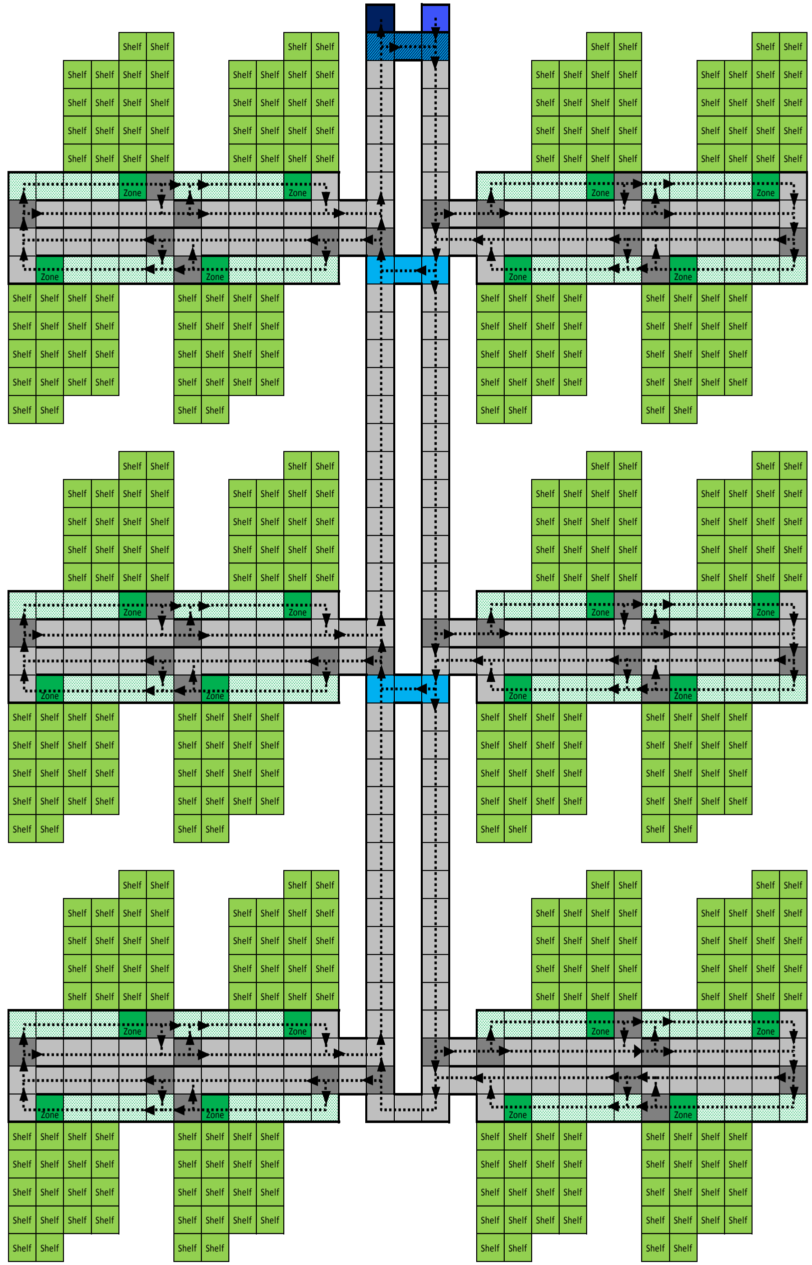


Effect of increasing the launch rate

- Initial scenario: launch rate = EXP (0.0167) totes/second
- New scenario: launch rate = EXP (0.03) totes/second
  - Constraint: models with recirculation and congestion
- **Effect:**
  - **Higher average throughput rate**
  - **Lower make span**
  - **Recirculation still increases average throughput time**






Shelf
Zone

Most efficient model:
12 zones, shortcuts, class-based storage, no recirculation