UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

A SIMULATION APPROACH TO ENHANCE WAREHOUSE LOGISTICS PERFORMANCE IN THE CERAMIC TILE SECTOR

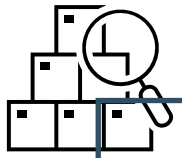
Marco Taccini[1], Giulia Dotti[2], Manuel Iori[1], Anand Subramanian[3]

marco.taccini@unimore.it

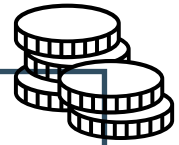
[1] Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy
[2] Department of Economics and Marco Biagi Foundation, University of Modena and Reggio Emilia, Italy
[3] Department of Computer Systems, Federal University of Paraíba, Brazil

THE CERAMIC TILE SECTOR

In 2021:



18.3 billion
of square meters
worldwide



€6.2 billion
of revenues
in Italy

ACIMAC Research Department 2022, October. **World Production and Consumption of Ceramic Tiles.** Technical Report 10, *Manufacturing Economics Study.*

Baraldi, L. 2022. **World Production and Consumption of Ceramic Tiles.** *Ceramic World Review* 148:36–57.

LITERATURE REVIEW

Production process:

- Xie, X. 2008. **A Review of Recent Advances in Surface Defect Detection Using Texture Analysis Techniques**. *Electronic Letters on Computer Vision and Image Analysis* 7(3):1–22.

Sustainability perspective:

- Medina-Salgado, M. S., F. E. García-Muiña, M. Cucchi, and D. Settembre-Blundo. 2021. **Adaptive Life Cycle Costing (LCC) Modeling and Applying to Italy Ceramic Tile Manufacturing Sector: Its Implication of Open Innovation**. *Journal of Open Innovation: Technology, Market, and Complexity* 7(1):101.

Decision Support Systems:

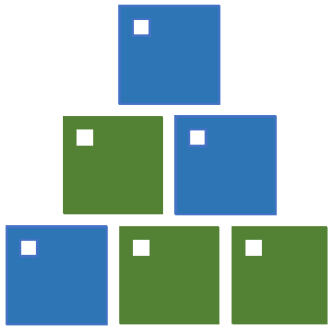
- Soares, A., C. Pimentel, and A. Moura. 2022. **Production Planning and Scheduling With Applications in the Tile Industry**. *International Journal of Operational Research* 44(1):58–79.

- Abdolazimi, O., D. Shishebori, F. Goodarzian, P. Ghasemi, and A. Appolloni. 2021. **Designing a New Mathematical Model Based on ABC Analysis for Inventory Control Problem: A Real Case Study**. *RAIRO-Operations Research* 55(4):2309–2335.

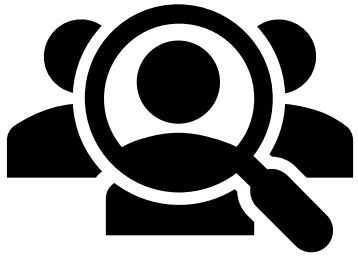
Simulation:

- Davoli, G., S. A. Gallo, M. Collins, and R. Melloni. 2010. **A Stochastic Simulation Approach for Production Scheduling and Investment Planning in the Tile Industry**. *International Journal of Engineering, Science and Technology* 2(9):107–124.

LACK OF HOMOGENEITY IN PRODUCTS



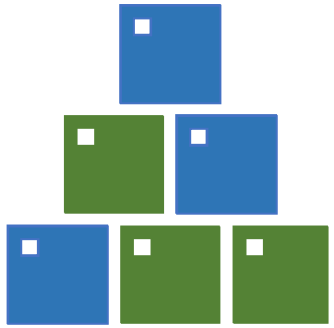
Lack of Homogeneity in the Product



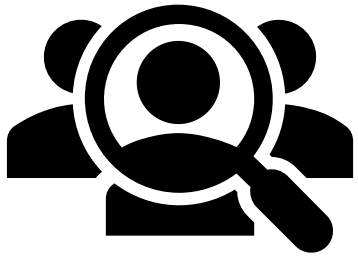
Customer demand

Boza, A., M. Alemany, F. Alarcón, and L. Cuenca. 2014. **A Model-Driven DSS Architecture for Delivery Management in Collaborative Supply Chains With Lack of Homogeneity in Products.** *Production Planning & Control* 25(8):650–661.

LACK OF HOMOGENEITY IN PRODUCTS

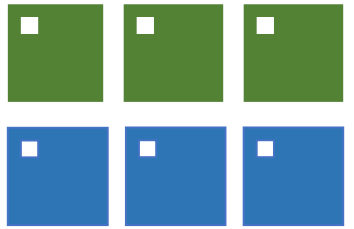


Lack of Homogeneity
in the Product



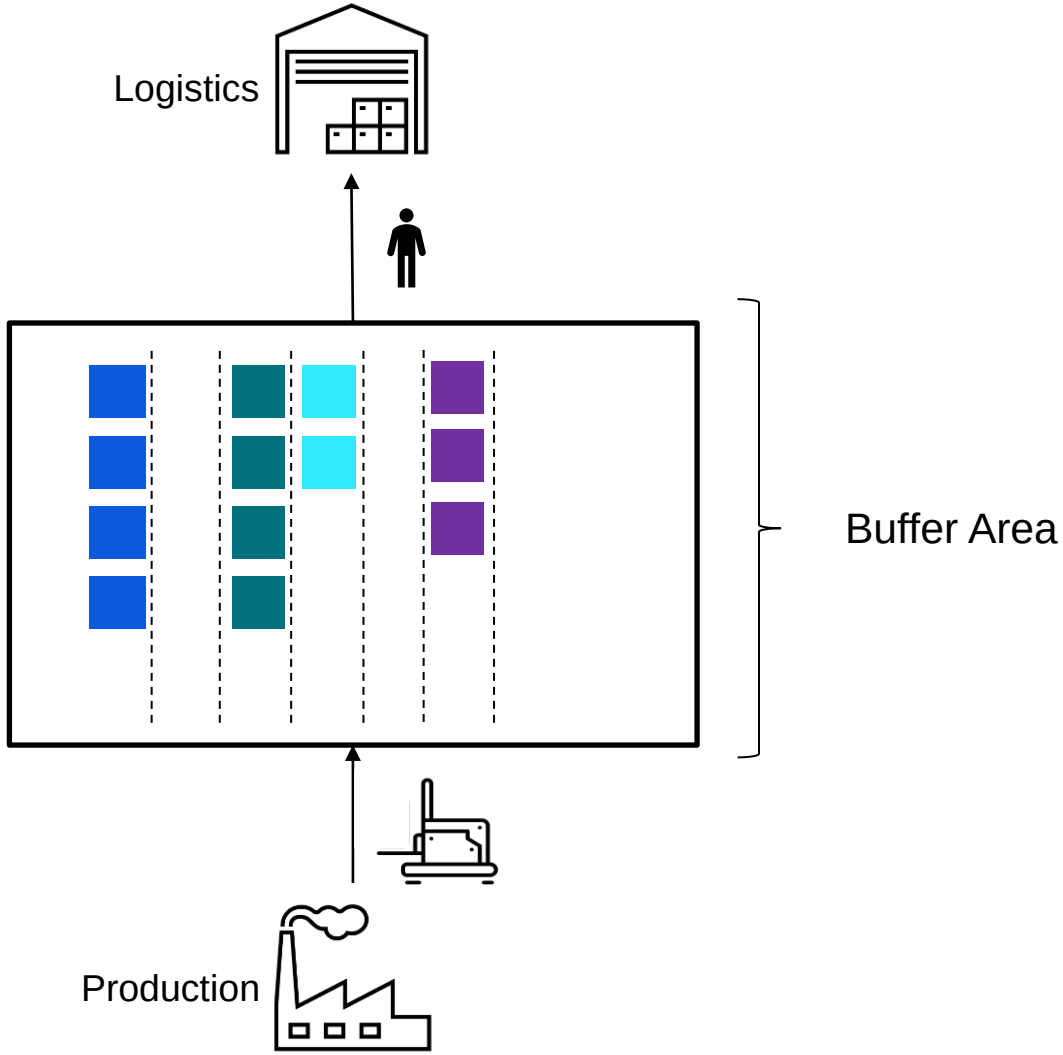
Customer demand

Visual Inspection
&
Classification Stage



Boza, A., M. Alemany, F. Alarcón, and L. Cuenca. 2014. **A Model-Driven DSS Architecture for Delivery Management in Collaborative Supply Chains With Lack of Homogeneity in Products.** *Production Planning & Control* 25(8):650–661.

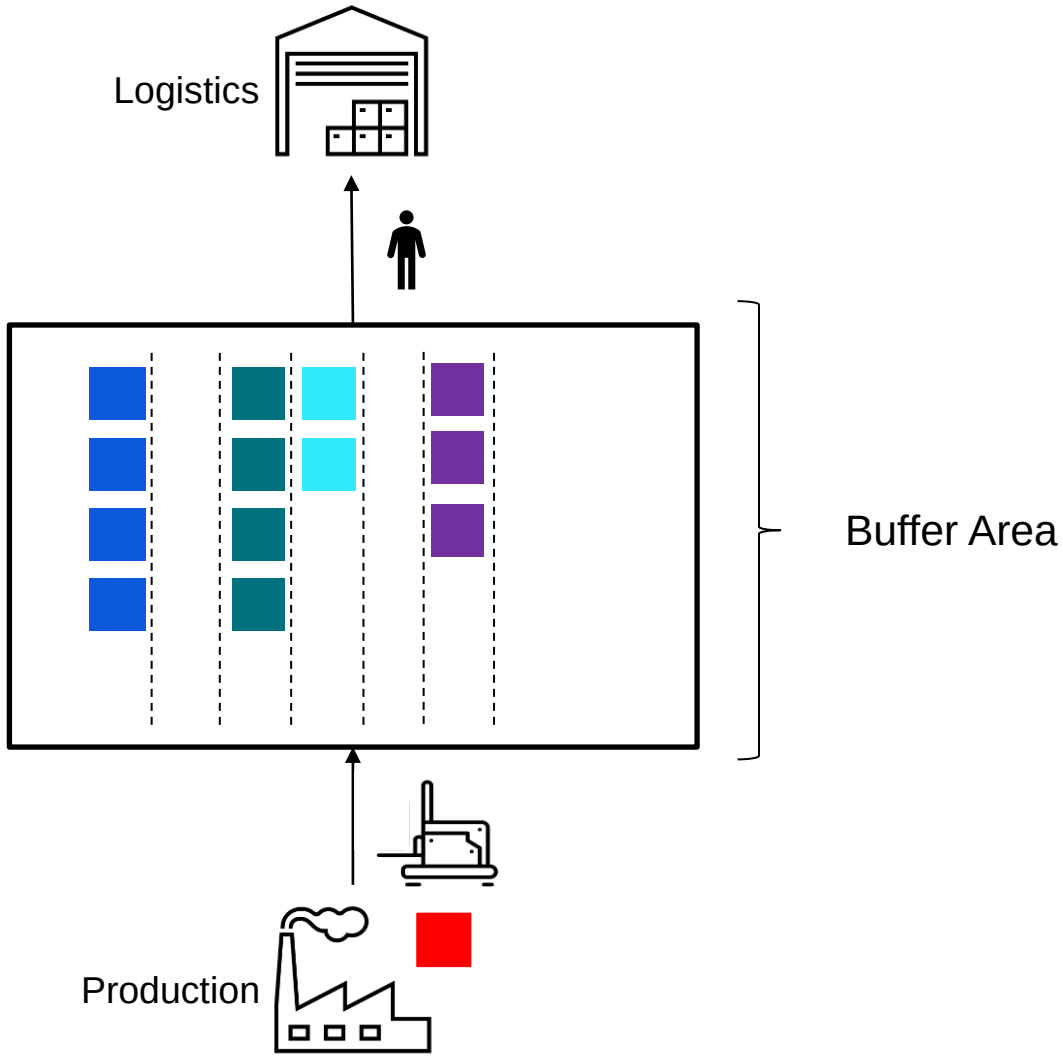
PROBLEM DESCRIPTION



How to select the best storage policy in order to:

- reduce costs;
- increase flexibility.

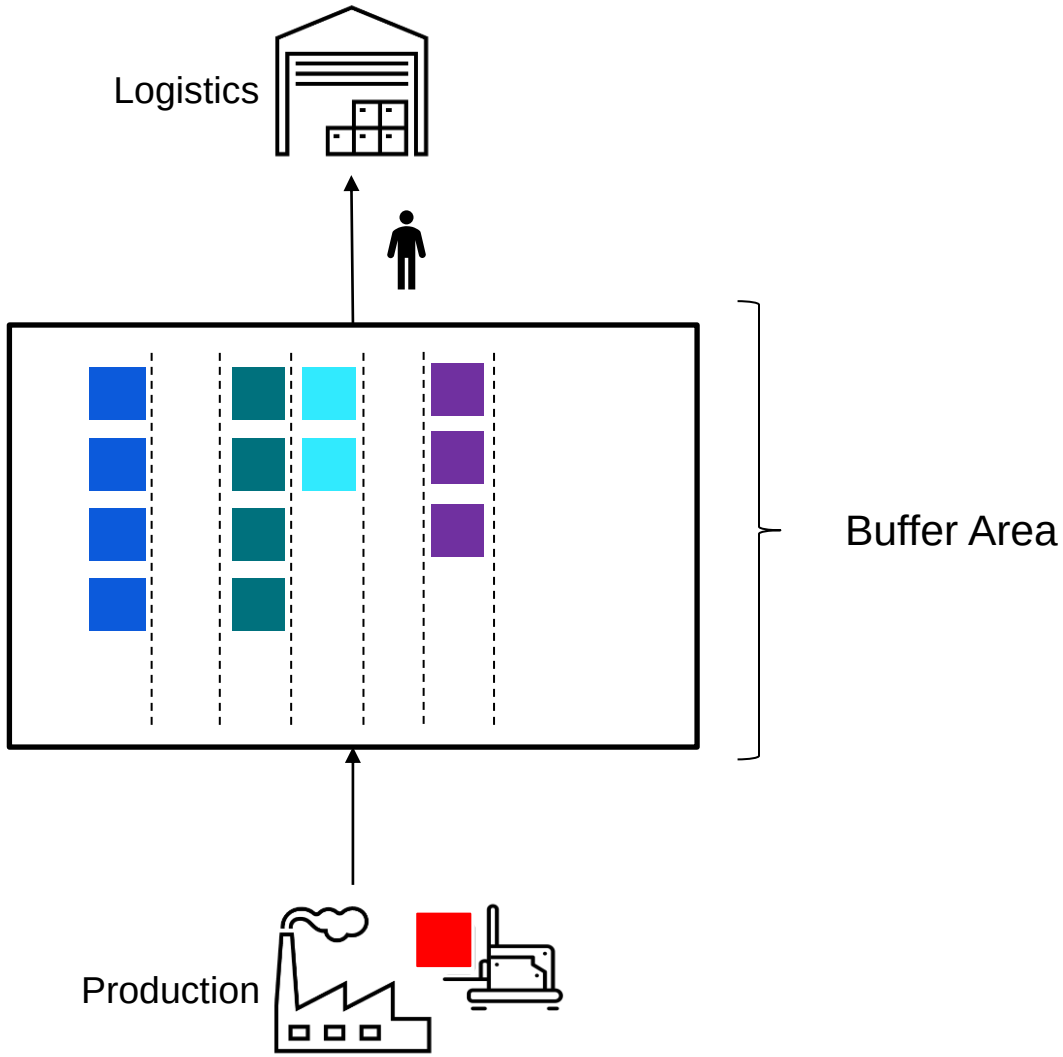
PROBLEM DESCRIPTION



How to select the best storage policy in order to:

- reduce costs;
- increase flexibility.

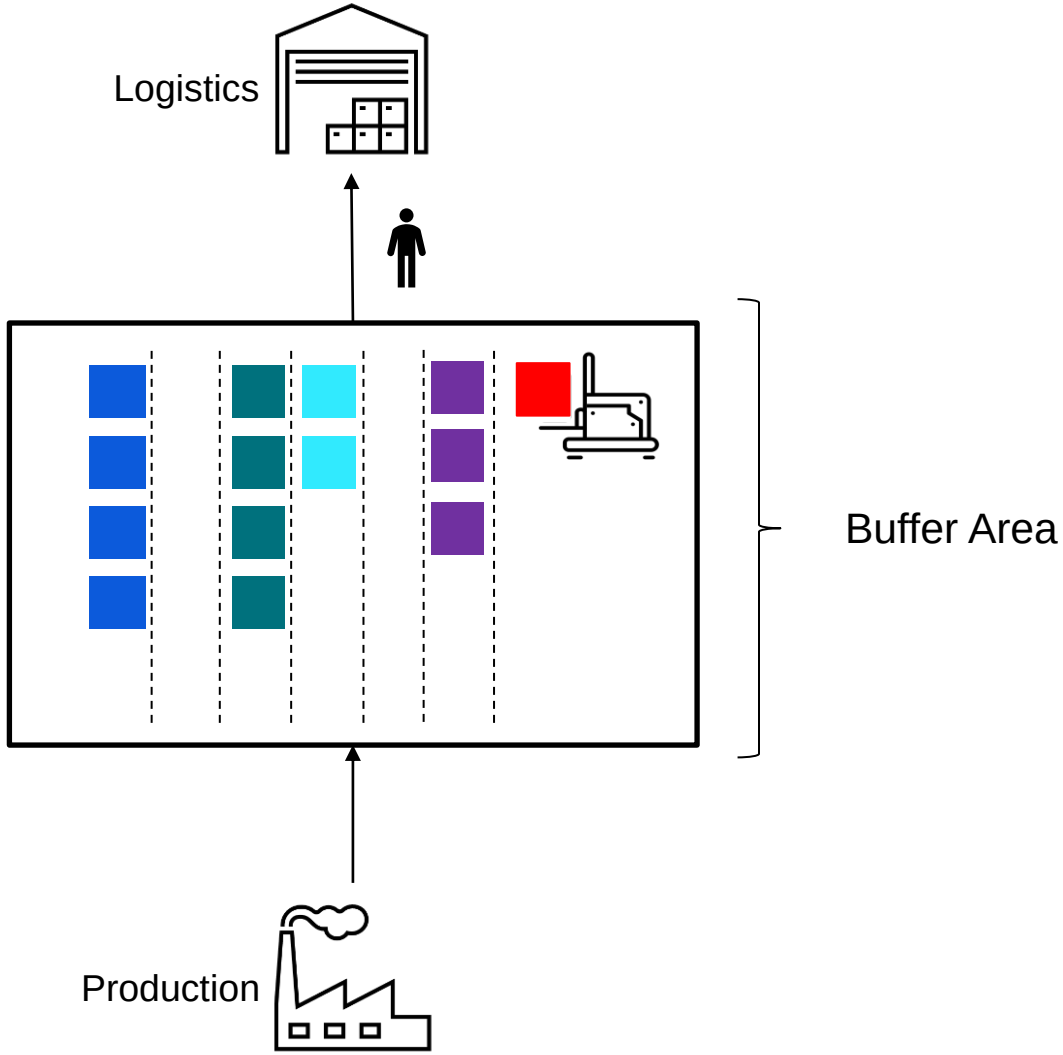
PROBLEM DESCRIPTION



How to select the best storage policy in order to:

- reduce costs;
- increase flexibility.

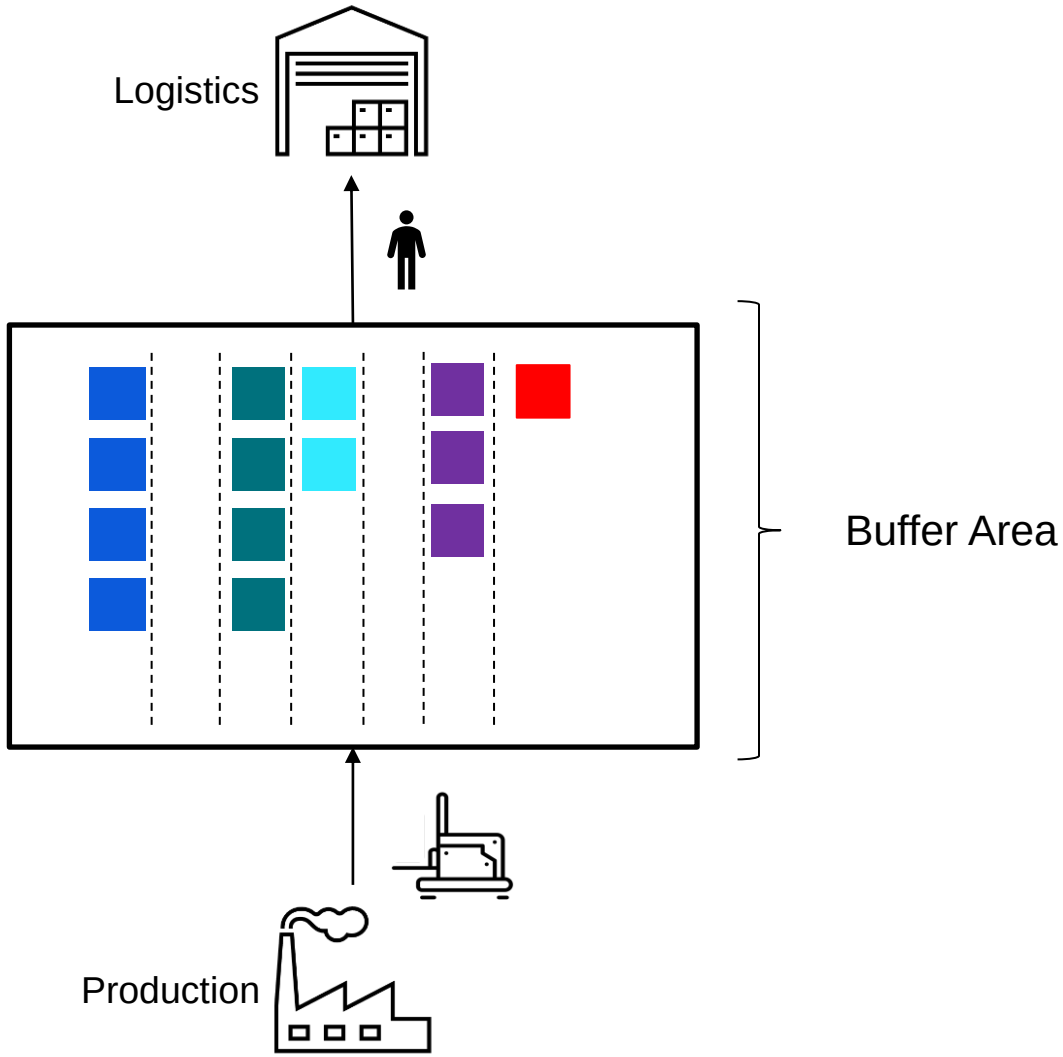
PROBLEM DESCRIPTION



How to select the best storage policy in order to:

- reduce costs;
- increase flexibility.

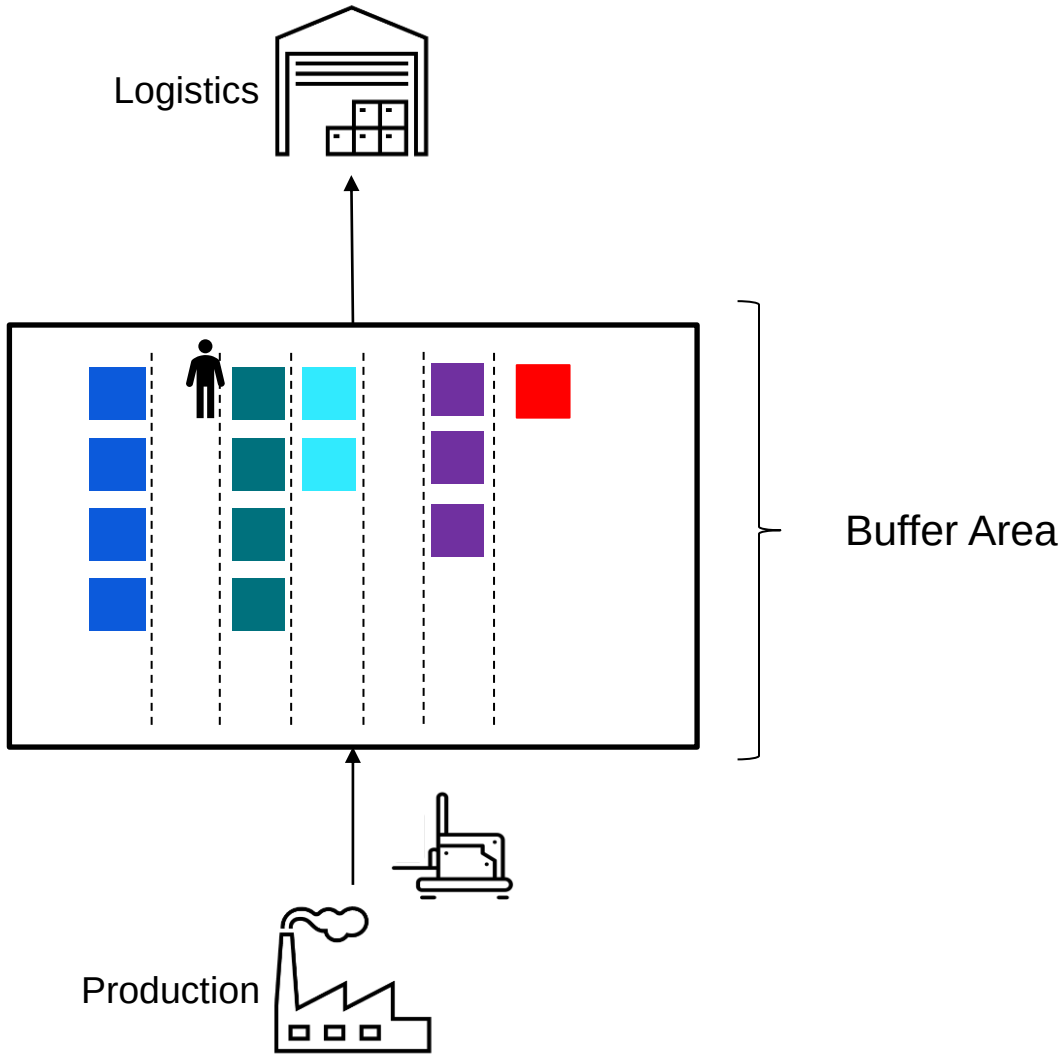
PROBLEM DESCRIPTION



How to select the best storage policy in order to:

- reduce costs;
- increase flexibility.

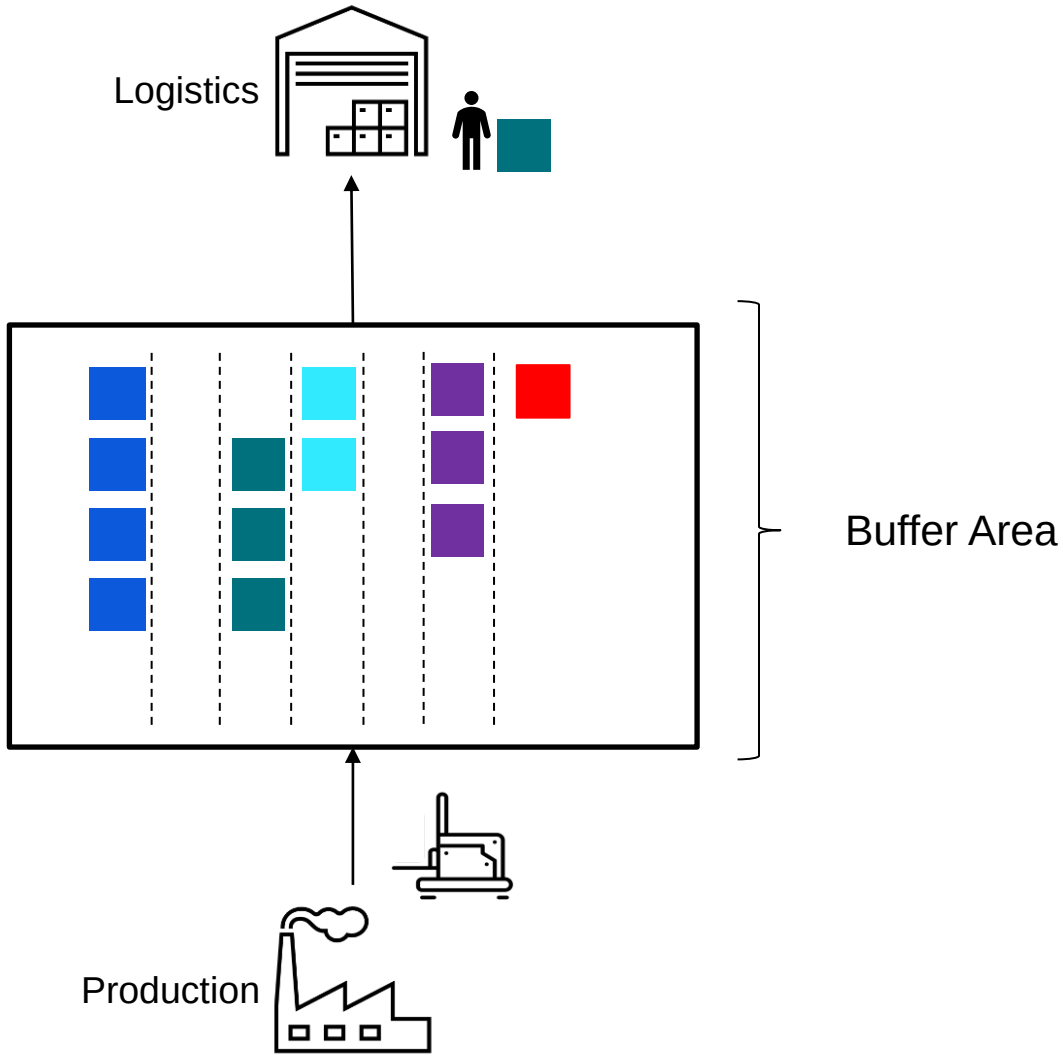
PROBLEM DESCRIPTION



How to select the best storage policy in order to:

- reduce costs;
- increase flexibility.

PROBLEM DESCRIPTION

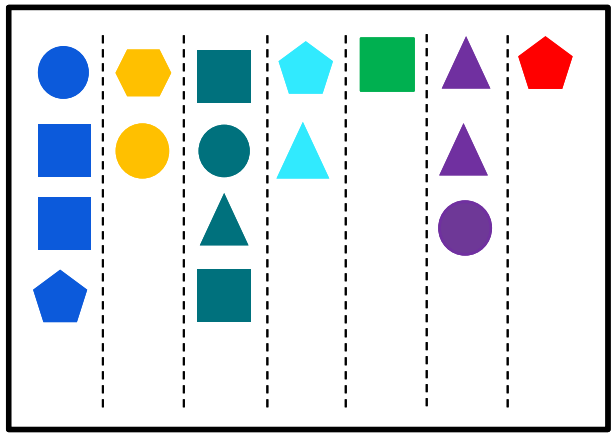


How to select the best storage policy in order to:

- reduce costs;
- increase flexibility.

STORAGE POLICIES

Current storage policy



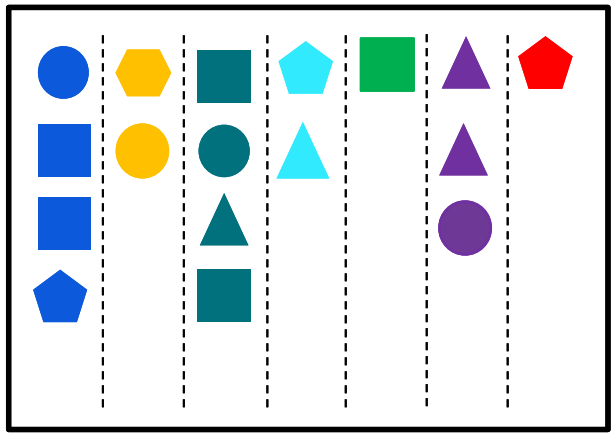
Buffer Area

Current classification based on **codes**:

- Technical specifications (shade and caliber);
- Commercial properties (size and tickness).

STORAGE POLICIES

Current storage policy

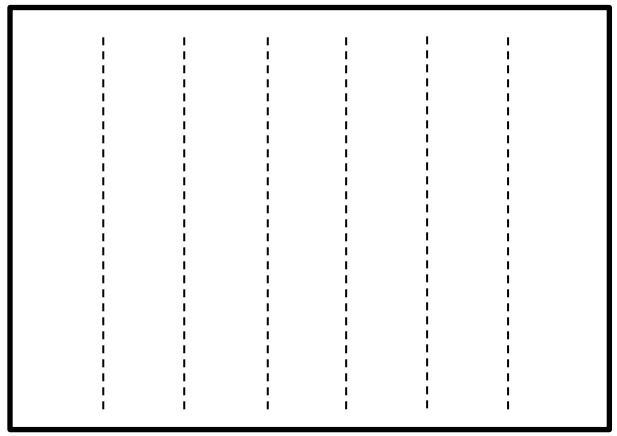


Buffer Area

Current classification based on **codes**:

- Technical specifications (shade and caliber);
- Commercial properties (size and tickness).

Proposed storage policy

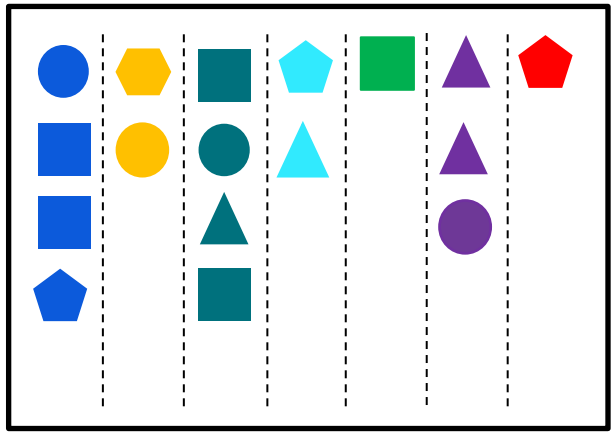


Buffer Area

Proposed classification based on **downstream destination**

STORAGE POLICIES

Current storage policy

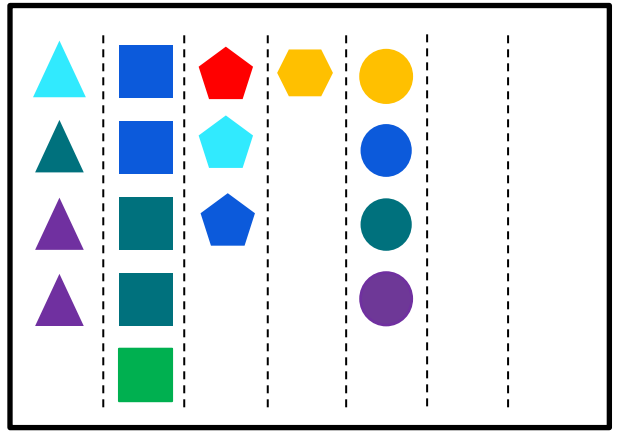


Buffer Area

Current classification based on **codes**:

- Technical specifications (shade and caliber);
- Commercial properties (size and tickness).

Proposed storage policy



Buffer Area

Proposed classification based on **downstream destination**

SALABIM

Useful functionalities:

- Open-source
- Developed in Python language
- Simula activate/passivate/hold paradigm
- Animations
- Queues
- Tracing
- Statistical distributions

van der Ham, R. 2018. **Salabim: Discrete Event Simulation and Animation in Python.** *Journal of Open Source Software* 3(27):767–768.

SALABIM – LITERATURE REVIEW

Scheduling:

- Lang, S., F. Behrendt, N. Lanzerath, T. Reggelin, and M. Müller. 2020. **Integration of Deep Reinforcement Learning and Discrete-Event Simulation for Real-Time Scheduling of a Flexible Job Shop Production.** In *Proceedings of the 2020 Winter Simulation Conference*, 3057–3068. Piscataway

- Lang, S., T. Reggelin, F. Behrendt, and A. Nahhas. 2020b. **Evolving Neural Networks to Solve a Two-Stage Hybrid Flow Shop Scheduling Problem with Family Setup Times.** *In Proceedings of the 2020 Hawaii International Conference on System Sciences,* 1298–1307.

- Anglano, C., M. Canonico, and M. Guazzone. 2019. **Online User-Driven Task Scheduling for FemtoClouds.** *In 2019 Fourth International Conference on Fog and Mobile Edge Computing*, 5–12.

Health Center Operations:

- Baldwa, V., S. Sehgal, V. Ramamohan, and V. Tandon. 2020. **A Combined Simulation and Machine Learning Approach for Real-Time Delay Prediction for Waitlisted Neurosurgery Candidates.** *In Proceedings of the 2020 Winter Simulation Conference*, 956–967.

- Shoaib, M., and V. Ramamohan. 2022. **Simulation Modeling and Analysis of Primary Health Center Operations.** *Simulation* 98(3):183–208.

SALABIM'S IMPLEMENTATION

PalletGeneratorClass

```
1: for pallet in input_data do
2:         wait until arrivaltime = sim.time
3:         Pallet(pallet)
4: end for
```

ForkliftClass

```
5: while True do
6:         Wait workingtime
7:         if bufferqueue is not empty then
8:                 pallet = getPalletToRemove()
9:                 Activate pallet
10:        else
11:                Passivate self
12:        end if
13: end while
```

PalletClass

```
14: x, y, z = storePolicy(self)
15: if x ≠ 1 and y ≠ 1 and z ≠ 1 then
16:         self.location = x, y, z
17: else
18:         return noSpace(self)
19: end if
20: Enter bufferqueue
21: Update animation
22: for forklift in forklifts do
23:         if forklift.state = Passive then
24:                 Activate forklift
25:         end if
26: end for
27: Passivate self
28: Leave bufferqueue
29: Update animation
```

SALABIM'S IMPLEMENTATION

1 Creation and Classification

PalletGeneratorClass

```
1: for pallet in input_data do
2:         wait until arrivaltime = sim.time
3:         Pallet(pallet)
4: end for
```

ForkliftClass

```
5: while True do
6:         Wait workingtime
7:         if bufferqueue is not empty then
8:                 pallet = getPalletToRemove()
9:                 Activate pallet
10:        else
11:                Passivate self
12:        end if
13: end while
```

PalletClass

```
14: x, y, z = storePolicy(self)
15: if x ≠ 1 and y ≠ 1 and z ≠ 1 then
16:         self.location = x, y, z
17: else
18:         return noSpace(self)
19: end if
20: Enter bufferqueue
21: Update animation
22: for forklift in forklifts do
23:         if forklift.state = Passive then
24:                 Activate forklift
25:         end if
26: end for
27: Passivate self
28: Leave bufferqueue
29: Update animation
```

SALABIM'S IMPLEMENTATION

2 Storage

PalletGeneratorClass

```
1: for pallet in input_data do
2:         wait until arrivaltime = sim.time
3:         Pallet(pallet)
4: end for
```

ForkliftClass

```
5: while True do
6:         Wait workingtime
7:         if bufferqueue is not empty then
8:                 pallet = getPalletToRemove()
9:                 Activate pallet
10:        else
11:                Passivate self
12:        end if
13: end while
```

PalletClass

```
14: x, y, z = storePolicy(self)
15: if x ≠ 1 and y ≠ 1 and z ≠ 1 then
16:         self.location = x, y, z
17: else
18:         return noSpace(self)
19: end if
20: Enter bufferqueue
21: Update animation
22: for forklift in forklifts do
23:         if forklift.state = Passive then
24:                 Activate forklift
25:         end if
26: end for
27: Passivate self
28: Leave bufferqueue
29: Update animation
```

SALABIM'S IMPLEMENTATION

```
PalletGeneratorClass
 1: for pallet in input_data do
 2:         wait until arrivaltime = sim.time
 3:         Pallet(pallet)
 4: end for

ForkliftClass
 5: while True do
 6:         Wait workingtime
 7:         if bufferqueue is not empty then
 8:                 pallet = getPalletToRemove()
 9:                 Activate pallet
10:         else
11:                 Passivate self
12:         end if
13: end while
```

```
PalletClass
14: x, y, z = storePolicy(self)
15: if x ≠ 1 and y ≠ 1 and z ≠ 1 then
16:         self.location = x, y, z
17: else
18:         return noSpace(self)
19: end if
20: Enter bufferqueue
21: Update animation
22: for forklift in forklifts do
23:         if forklift.state = Passive then
24:                 Activate forklift
25:         end if
26: end for
27: Passivate self
28: Leave bufferqueue
29: Update animation
```

3 Queueing

SALABIM'S IMPLEMENTATION

4 Pick-up

PalletGeneratorClass

```
1: for pallet in input_data do
2:         wait until arrivaltime = sim.time
3:         Pallet(pallet)
4: end for
```

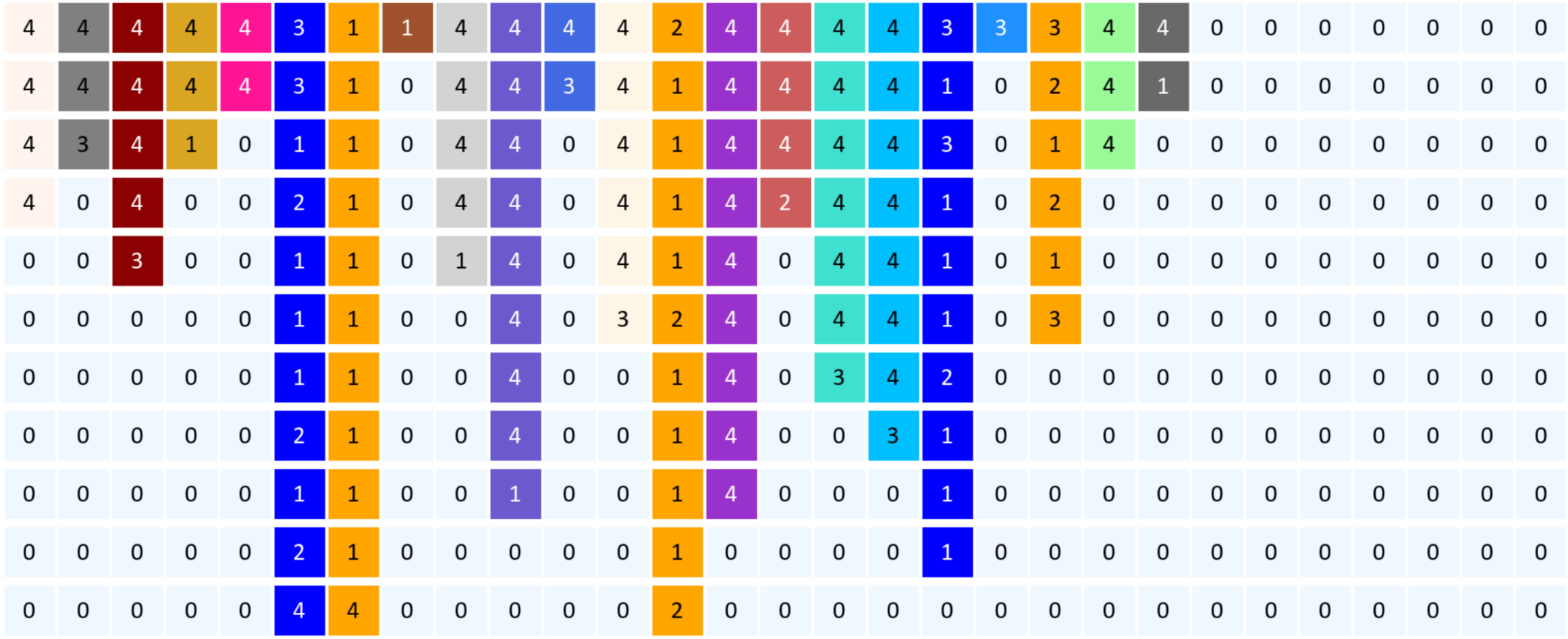
ForkliftClass

```
5: while True do
6:         Wait workingtime
7:         if bufferqueue is not empty then
8:                 pallet = getPalletToRemove()
9:                 Activate pallet
10:        else
11:                Passivate self
12:        end if
13: end while
```

PalletClass

```
14: x, y, z = storePolicy(self)
15: if x ≠ 1 and y ≠ 1 and z ≠ 1 then
16:         self.location = x, y, z
17: else
18:         return noSpace(self)
19: end if
20: Enter bufferqueue
21: Update animation
22: for forklift in forklifts do
23:         if forklift.state = Passive then
24:                 Activate forklift
25:         end if
26: end for
27: Passivate self
28: Leave bufferqueue
29: Update animation
```

VERIFICATION AND VALIDATION



PERFORMANCE INDICATORS

Without operators

Time before occurrence of mixed columns [minutes]

Time before area being filled [minutes]

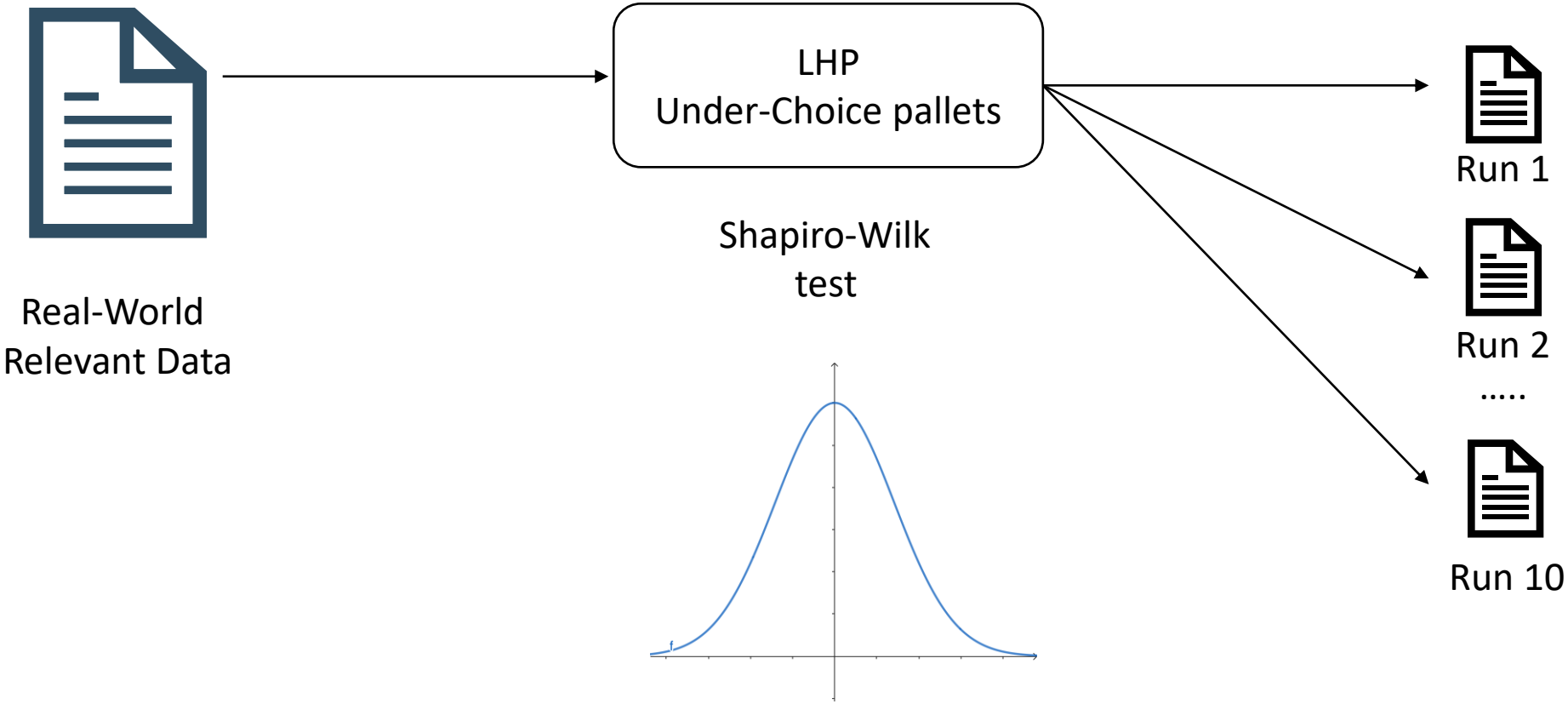
Number of lost pallets

With operators

Maximum number of columns used simultaneously

INPUT DATA

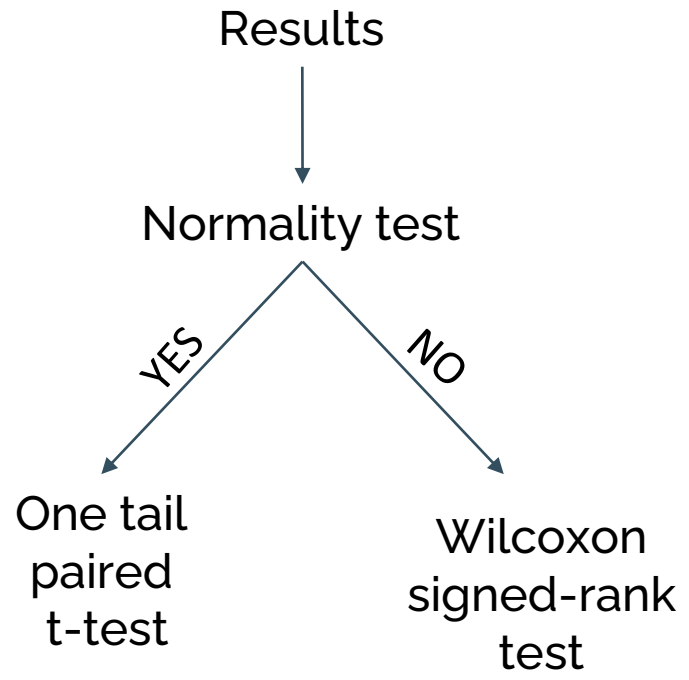
30 Working Days Simulated



RESULTS

	Without operators						With operators	
	Minutes before *mix/soft-mix* state		Minutes before area filled		Number of lost pallets		Max. number of columns	
Day	Current	Proposed	Current	Proposed	Current	Proposed	Current	Proposed
1	634.2	774.1	1322.8	1334.7	57.7	55.1	26.0	23.5
2	660.7	779.6	1297.2	1469.6	45.9	0.0	28.5	24.2
3	388.3	486.0	1229.0	1177.8	80.4	108.0	29.0	29.0
4	511.4	695.3	1174.4	1230.9	113.2	68.9	28.3	25.1
5	790.9	1019.2	1391.7	1432.2	43.6	3.2	22.2	16.2
6	453.0	681.7	1280.3	1319.1	83.6	53.9	28.8	25.9
7	776.1	966.9	1421.7	1495.9	0.0	0.0	21.3	17.1
8	527.0	729.4	1395.9	1367.5	14.3	20.7	26.9	23.7
9	602.3	851.0	1382.6	1409.9	32.9	15.8	26.3	22.0
10	881.9	948.7	1474.1	1429.8	0.8	2.5	22.3	18.6
11	600.2	718.9	1311.6	1364.3	49.6	31.5	26.8	23.4
12	759.0	847.9	1467.7	1467.6	0.0	0.0	23.6	20.7
13	715.1	972.9	1305.4	1378.5	49.8	22.6	23.7	19.1
14	618.1	887.4	1309.7	1451.8	61.6	1.5	26.8	21.8
15	764.1	803.0	1370.1	1402.7	16.0	4.8	26.2	21.7
16	481.8	694.8	1356.2	1424.7	55.8	14.7	28.8	24.6
17	747.1	930.6	1361.7	1442.0	34.0	1.0	24.7	21.1
18	601.3	782.8	1327.7	1256.4	0.0	0.0	18.8	17.3
19	530.5	913.7	1262.4	1435.5	85.5	2.7	28.3	23.1
20	1004.8	1099.0	1524.0	1498.4	0.0	2.5	22.2	18.7
21	539.5	724.8	1203.5	1245.6	79.5	70.7	28.1	22.9
22	580.0	758.4	1454.7	1419.5	2.7	4.0	26.2	22.4
23	430.6	534.5	1109.4	1118.6	195.5	203.8	29.0	28.6
24	699.6	910.9	1367.9	1476.5	22.6	1.0	25.6	22.1
25	620.1	836.1	1361.3	1477.7	17.7	0.0	26.4	23.6
26	780.6	1178.9	1488.3	1595.0	3.2	0.0	23.1	19.8
27	646.8	835.1	1368.1	1392.4	43.0	37.5	25.6	20.9
28	734.3	937.4	1525.7	1408.2	0.0	9.5	22.2	18.1
29	878.3	928.0	1436.8	1399.6	3.4	35.5	23.2	20.6
30	385.1	511.5	1157.3	1169.2	150.6	157.9	29.0	28.2

Proven statistical significant difference



RESULTS

	Without operators						With operators	
	Minutes before *mix/soft-mix* state		Minutes before area filled		Number of lost pallets		Max. number of columns	
Day	Current	Proposed	Current	Proposed	Current	Proposed	Current	Proposed
1	634.2	774.1	1322.8	1334.7	57.7	55.1	26.0	23.5
2	660.7	779.6	1297.2	1469.6	45.9	0.0	28.5	24.2
3	388.3	486.0	1229.0	1177.8	80.4	108.0	29.0	29.0
4	511.4	695.3	1174.4	1230.9	113.2	68.9	28.3	25.1
5	790.9	1019.2	1391.7	1432.2	43.6	3.2	22.2	16.2
6	453.0	681.7	1280.3	1319.1	83.6	53.9	28.8	25.9
7	776.1	966.9	1421.7	1495.9	0.0	0.0	21.3	17.1
8	527.0	729.4	1395.9	1367.5	14.3	20.7	26.9	23.7
9	602.3	851.0	1382.6	1409.9	32.9	15.8	26.3	22.0
10	881.9	948.7	1474.1	1429.8	0.8	2.5	22.3	18.6
11	600.2	718.9	1311.6	1364.3	49.6	31.5	26.8	23.4
12	759.0	847.9	1467.7	1467.6	0.0	0.0	23.6	20.7
13	715.1	972.9	1305.4	1378.5	49.8	22.6	23.7	19.1
14	618.1	887.4	1309.7	1451.8	61.6	1.5	26.8	21.8
15	764.1	803.0	1370.1	1402.7	16.0	4.8	26.2	21.7
16	481.8	694.8	1356.2	1424.7	55.8	14.7	28.8	24.6
17	747.1	930.6	1361.7	1442.0	34.0	1.0	24.7	21.1
18	601.3	782.8	1327.7	1256.4	0.0	0.0	18.8	17.3
19	530.5	913.7	1262.4	1435.5	85.5	2.7	28.3	23.1
20	1004.8	1099.0	1524.0	1498.4	0.0	2.5	22.2	18.7
21	539.5	724.8	1203.5	1245.6	79.5	70.7	28.1	22.9
22	580.0	758.4	1454.7	1419.5	2.7	4.0	26.2	22.4
23	430.6	534.5	1109.4	1118.6	195.5	203.8	29.0	28.6
24	699.6	910.9	1367.9	1476.5	22.6	1.0	25.6	22.1
25	620.1	836.1	1361.3	1477.7	17.7	0.0	26.4	23.6
26	780.6	1178.9	1488.3	1595.0	3.2	0.0	23.1	19.8
27	646.8	835.1	1368.1	1392.4	43.0	37.5	25.6	20.9
28	734.3	937.4	1525.7	1408.2	0.0	9.5	22.2	18.1
29	878.3	928.0	1436.8	1399.6	3.4	35.5	23.2	20.6
30	385.1	511.5	1157.3	1169.2	150.6	157.9	29.0	28.2
	27.9 %		2.6 %		30.8 %		13.5 %	

Without operators:

1) Positive impact on worker activities, leading to reduced operation time

2) Capacity improvements and reduction in production downtimes

3) Reduction of economic losses

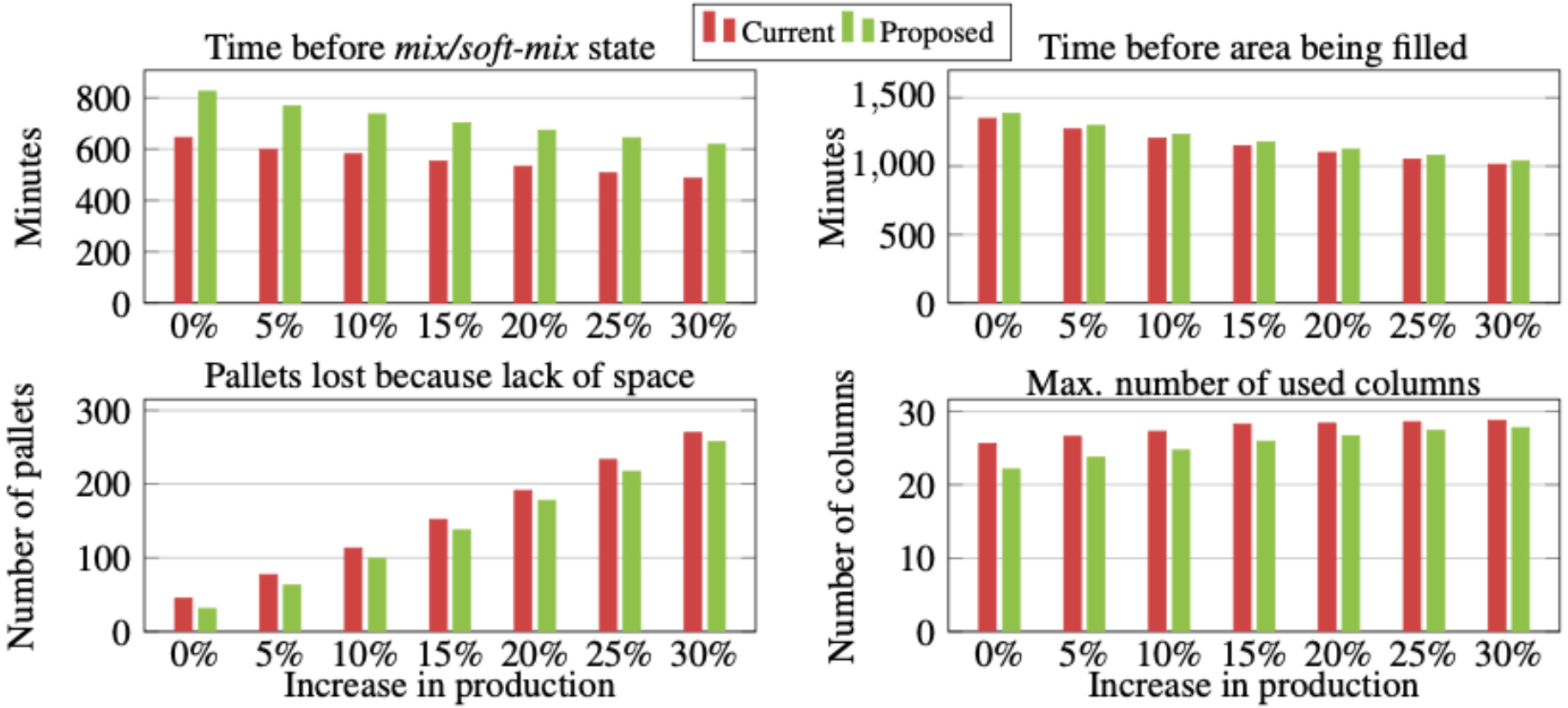
With operators:

1) Reduction of the buffer area

SENSITIVITY ANALYSIS

SEVEN SCENARIOS:

- +0% production
- +5% production
- +10% production
- +15% production
- +20% production
- +25% production
- +30% production



CONCLUSIONS

- ✓ Overview of the characteristics of the ceramic sector
- ✓ Policies (characteristics-based and destination-based)
- ✓ Comparative statistical analysis using Discrete Event Simulation
- ✓ Sensitivity analysis of increased production scenarios

- Real-world implementation
- Accurate representation of the process -> continuous improvement

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

A SIMULATION APPROACH TO ENHANCE WAREHOUSE LOGISTICS PERFORMANCE IN THE CERAMIC TILE SECTOR

Marco Taccini[1], Giulia Dotti[2], Manuel Iori[1], Anand Subramanian[3]

marco.taccini@unimore.it

[1] Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy
[2] Department of Economics and Marco Biagi Foundation, University of Modena and Reggio Emilia, Italy
[3] Department of Computer Systems, Federal University of Paraíba, Brazil