Assignment 1

Advanced modeling for operations
Professor: Elena Tappia
A.Y. 2024-2025

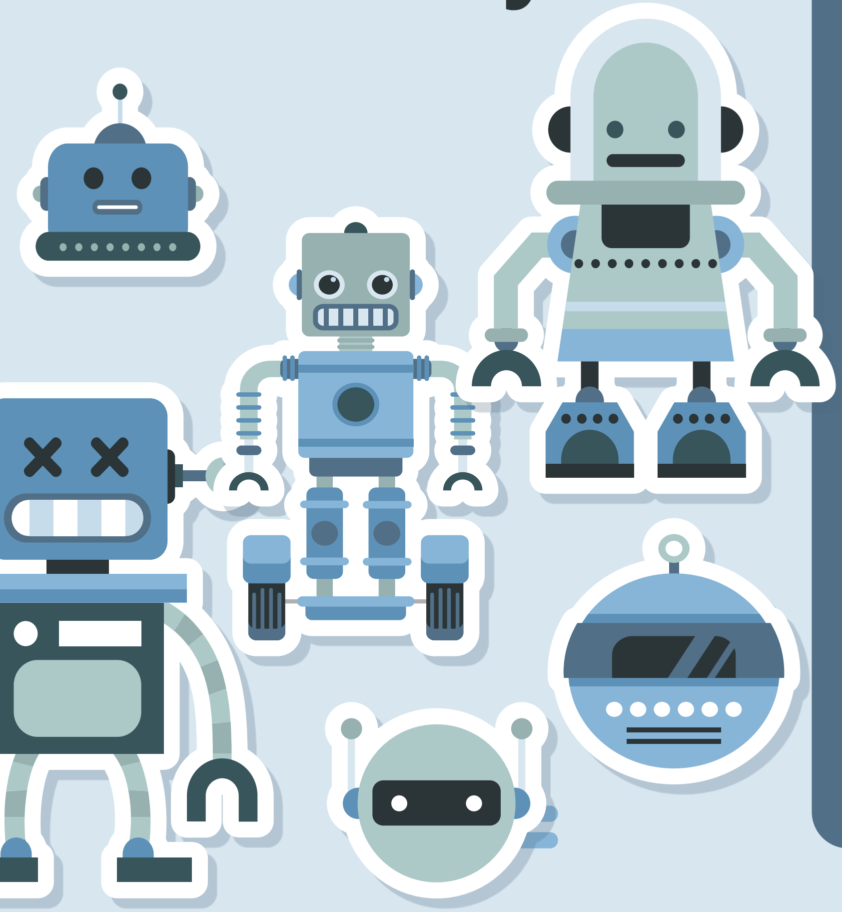
Our team

Matteo Schiattarella
10653754

Mattia Longobardo
10616746

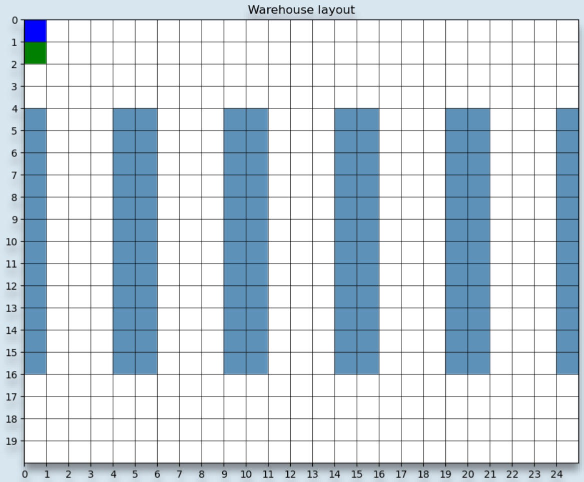
Carlo Andrea Russo
10669963

Summary



Base case

Objective: improve the performance of a hybrid picking system, where pickers and AMR work together collaboratively.



Warehouse layout

Picking aisles: 5 (each 12 metres long)
Cross aisles: 2 (each 4 metres long)
Total picking position: 480 (4 for each rack)
Input/output position: in the corner

Item characteristics

Weight: uniform distribution(0.1 – 2 Kg)
Volume: uniform distribution (0.1 – 1 dm^3)

Pickers and AMRs

Distances: Manhattan
Fixed AMRs speed: 1 m/s
Avg. picker speed: 0.95 m/s
Pickers and AMRs' number: 2 (both)
Pickers and AMRs' capacity: 4 (both)

Customer order

There are 250 orders, all known at the beginning of the picking wave. Processed by following a FCFS (First-Come-First-Served) policy and assigned to the first available picker and AMR

To be case

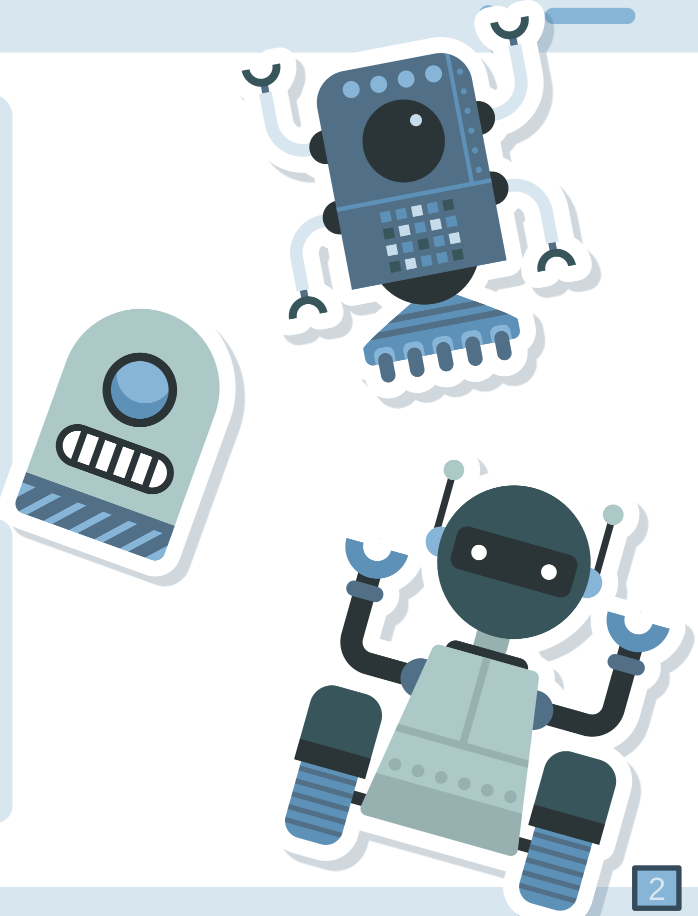
Improving rules

To develop the to be case was possible to improve rules in terms of:

- picking strategy (implementing the batch strategy)
- storage strategy (implementing class base strategy)
- order assignment strategy (implementing the order assignment driven by productivity or worker satisfaction)

Strategies analysis:

In order to carry out the most accurate analysis possible, for each of the strategies, searches were conducted in the existing literature, which led to the identification of several possible system configurations.



Additional Hypotheses

- Pickers don’t unload at I/O; only AMRs do

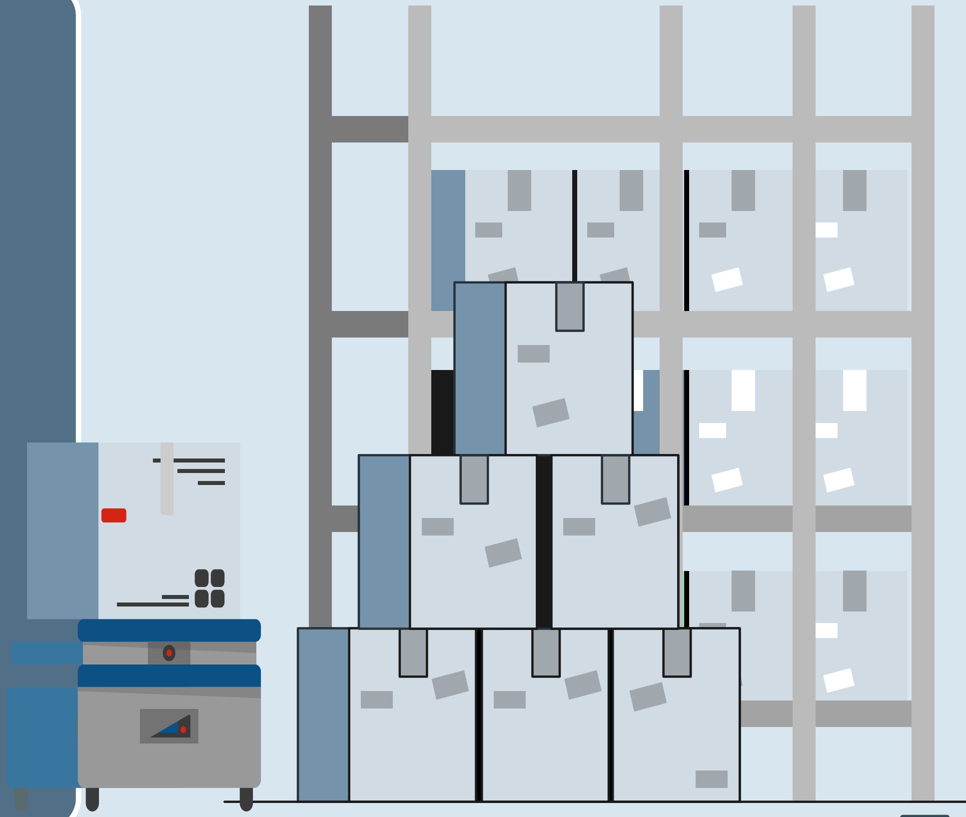
- The input/output is fixed (we assumed that relocating it is not possible, for example, due to high costs or other reasons)

-The maximum weight a picker can carry is $\mu + 1.5 * \omega$

-The maximum volume a picker can carry is $\mu + 1.5 * \omega$

-The order volume in an 8-hour shift is 250.

- Order assignment is included in the batching policies. So, some batching policies are designed to assign a balanced workload and others a more productive workload. For this reason, we also test 1 as batching size to highlight the effect of the order assignment.



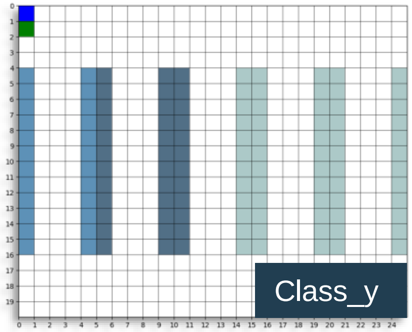
Storage strategies

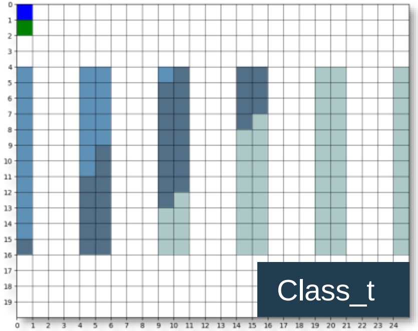
Random storage: the unit load can be stored in each pallet location, if available

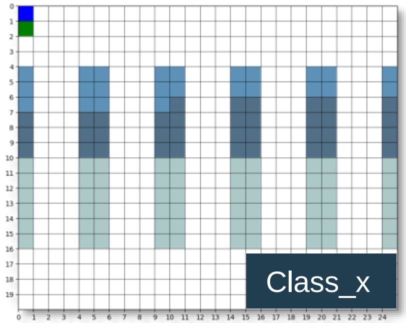
Class-based storage: The unit load of a specific family of items must be stored in a specific set of pallet locations. The family with the highest Access Index (AI) must be stored in the pallet locations closest to the I/O.

$$AI = \frac{Retriving\ index\ (\#\ of\ unit\ loads\ retrieved\ from\ the\ storage\ system\ during\ T)}{\#\ locations\ assigned\ during\ T}$$

According to the literature, moving from random storage to class-based storage improves the operative cycle time but increases the management complexity. After analysing the warehouse layout and the distribution of different items, three possible smart class zone shapes have been identified.







Legend:
- Class A
- Class B
- Class C

Batching strategies

*This type of batching reaches the maximum batch size or weight/volume, so it's not guaranteed that all batches will have the same batch size.
**This type of batching can also be seen as object similarity grouping because if 2 orders have the same object, they will also have the same xy similarity.

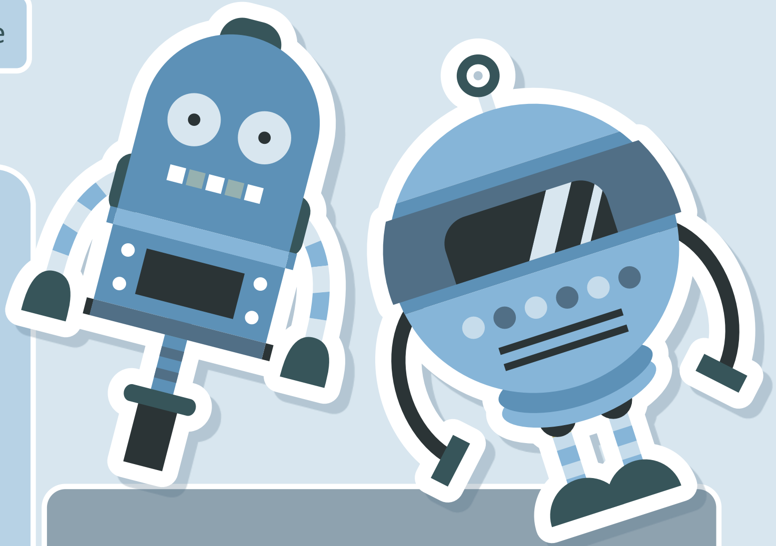
Random storage: the unit load can be stored in each pallet location, if available

Batch picking: the operator has to fulfil more than one order per mission

Usually, the batch size is defined as the value that optimizes the sorting and picking costs. There are no data about costs, so 4 values of size (1,2,3,4) have been tested with the aim of optimising time.

Furthermore, some possible solution for batch type have been developed on the basis of three drivers: time reduction, volume and weight (to satisfy productivity or operator satisfaction).

- **Random,** group items according to the order of the list of orders.
- **Volume,** group items by increasing volume
- **Volume-balanced,** orders are created with a volume limit set as μ+1.5·std.*
- **Weight,** group items by increasing weight (first sort by weight, then create groups).
- **Weight-balanced,** orders are created with a weight limit set as μ+1.5·std.*
- **XY similarity,** group items based on similarity along both the X and Y axes.**
- **Z similarity,** group items based on similarity in height.
- **Proximity**, Batch orders based on the aisles their items are located in. Orders with picks in the same or nearby aisles are grouped together using hierarchical clustering on aisle overlap



According to the literature, using a batching increase the project complexity because sorting is required but improve the picking time because of increase the picking density, reduce the average number of lines on the picking list and reduce the number of mission.

Routing strategies

Routing policies indicate the sequence according to which the items are picked and there are several possibilities.

Transversal policy: The picker enters the aisle, walks through it, stops when requested, and walks out of the other side.

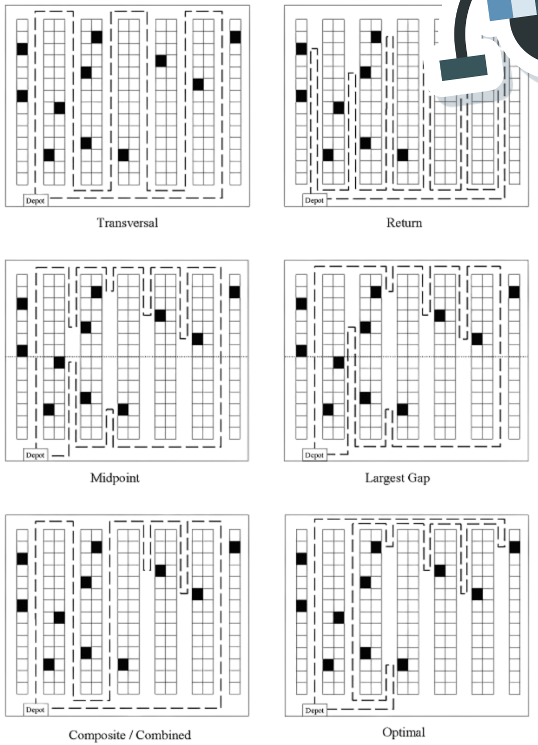
Return policy: The picker enters the aisle from one side, and after the picking activities, he goes out from the same side.

Mid-point return policy: the picking area is divided into 2 zones, and the picker adopts a return policy in both areas.

Largest gap return policy: for each aisle, the most significant gap is determined as the maximum distance between the beginning of the aisle to the first picking location, the distance between each picking location and the next one and the distance between the last picking location and the end of the aisle

Composite: for each aisle, the picker chooses the best policy between traversal and return policy

Optimal: The simplified algorithm for the travelling salesman problem defines the path.



Order assignment strategies

Productivity

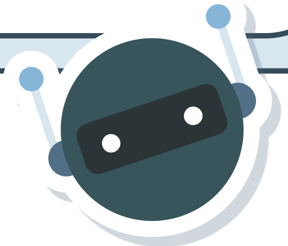
To measure the level of productivity, two dimensions have been considered: simulation time and number of simulation steps (the lower these values, the higher the productivity is).

Workers Satisfaction

To analyse the level of worker satisfaction 2 solutions based on 4 drivers (working time, volume, weight and workers steps) have been analysed.

Equal work distribution: Different workers have the same working time, volume, weight transported and number of steps for each simulation.

Alternate heavy work: The level of work is not equally distributed between the different workers, each worker alternate heavy and light days.



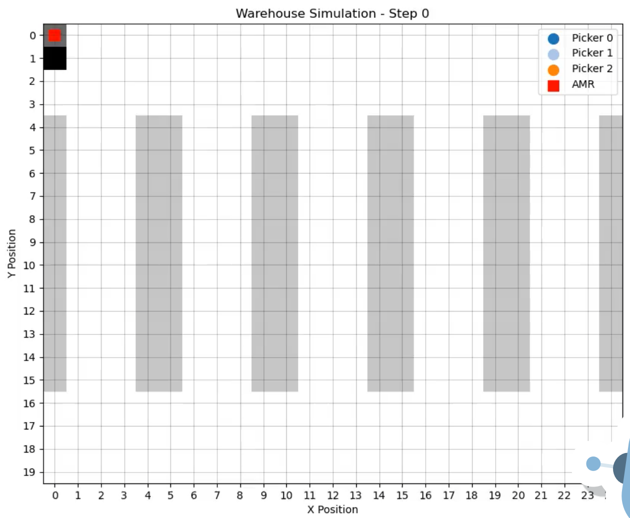
Other considerations:

- If a worker transport more than 200kg/h he is authorized to make a 5 minutes break.
- Free AMR is always assigned to the nearest picking point.
- The possibility of introducing another worker and/or another AMR was considered (#AMRs >= #Pickers)

Simulations

To analyse all the possible cases, 55,440 simulations were run.

The total time required was 37.25 minutes. 32 simulations were run simultaneously, with an average of 1.29 seconds per simulation.



The best case results in a **78.9%** improvement in simulation time (and the number of steps), reducing the time from 223 minutes to just 47 minutes.

The best case with 2 pickers,2 AMRs and AMR capacity equal to 4 reduced the time from 223 minutes to 75 minutes with an improvement of **65.9%**

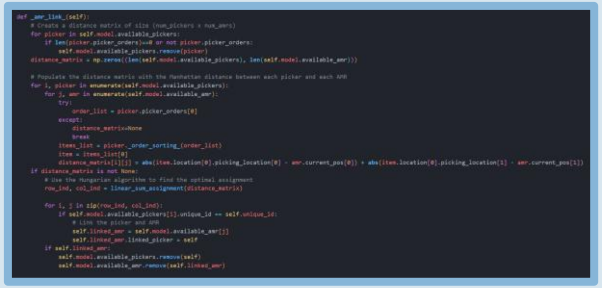
Code specifications

A* Algorithm:

The algorithm implements the A* pathfinding method to navigate a warehouse grid. It calculates the Manhattan distance as the heuristic and combines it with the movement cost to determine the most efficient path. A priority queue explores nodes with the lowest cost while respecting dynamic movement constraints and avoiding blocked aisles.

Smart AMR:

The smart AMR feature allows the robot to detach from the linked picker after unloading its load and to be reassigned to the optimal picker based on the Manhattan distance. Moreover, it ensures that the AMR's capacity is taken into consideration before proceeding to the unloading area.



```
def _amr_link_(self):
    # Create a distance matrix of size (num_pickers x num_amrs)
    for picker in self.model.available_pickers:
        if len(picker.picker_orders)==0 or not picker.picker_orders:
            self.model.available_pickers.remove(picker)
    distance_matrix = np.zeros((len(self.model.available_pickers), len(self.model.available_amr)))

    # Populate the distance matrix with the Manhattan distance between each picker and each AMR
    for i, picker in enumerate(self.model.available_pickers):
        for j, amr in enumerate(self.model.available_amr):
            try:
                order_list = picker.picker_orders[0]
            except:
                distance_matrix=None
                break
            items_list = picker._order_sorting_(order_list)
            item = items_list[0]
            distance_matrix[i][j] = abs(item.location[0].picking_location[0] - amr.current_pos[0]) + abs(item.location[0].picking_location[1] - amr.current_pos[1])
    if distance_matrix is not None:
        # Use the Hungarian algorithm to find the optimal assignment
        row_ind, col_ind = linear_sum_assignment(distance_matrix)

        for i, j in zip(row_ind, col_ind):
            if self.model.available_pickers[i].unique_id == self.unique_id:
                # Link the picker and AMR
                self.linked_amr = self.model.available_amr[j]
                self.linked_amr.linked_picker = self
        if self.linked_amr:
            self.model.available_pickers.remove(self)
            self.model.available_amr.remove(self.linked_amr)
```

Jaccard distances and hierarchical clustering:

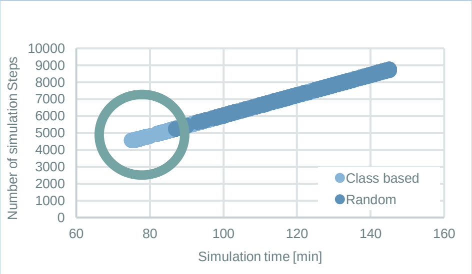
The proximity batches orders by representing each order as a binary vector indicating the aisles involved. It computes pairwise Jaccard distances between these vectors to measure aisle overlap between orders. Hierarchical clustering is then performed on these distances to group similar orders together, with the number of clusters determined by the batch size. The clustered orders are assigned to pickers in a balanced way, ensuring that each picker gets orders with nearby aisle locations. This approach integrates multiple techniques, including vectorisation, Jaccard distance calculation, hierarchical clustering, and balanced cluster assignment, to optimise task distribution and improve efficiency.

Productivity analysis

*All graph on this page are drawn for the scenario with 2 pickers, 2 AMRs, and AMR capacity of 4. However, the graphs are similar also changing the number of pickers, number of AMRs, or AMR capacity except for Smart AMRs graph when capacity is bigger than 4. (see excel file)

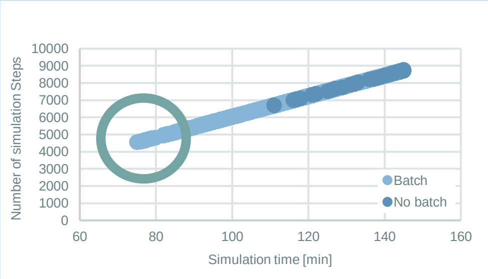
It is possible to separately observe the impact of introducing class-based, batch, and Smart AMRs policies. Specifically, the analysis includes both the baseline case with 2 pickers, 2 AMRs, and an AMR capacity of 4 (2p2A4c), as well as other notable configurations: 2p3a4c, 3p3a4c, 2p2aXc, and 3p3aXc.

Class based impact



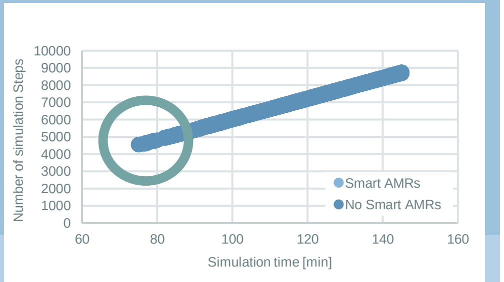
The introduction of class-based brings to a reduction of simulation time (and number of steps). Comparing the best case of class-based storage with the best case of random storage for all pickers and AMRs configurations there is always an improvement ranging between **13,8%** and **17,5%**

Batch impact

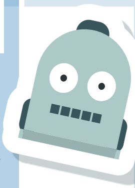


The introduction of batch results in a big reduction in simulation time (and number of steps). Comparing the best batch case with the best non-batch case leads to a reduction of simulation time ranging of **32,4%** and **35,0%** for all pickers and AMRs configurations.

Smart AMR impact



Introducing smart AMRs brings to a small reduction of simulation time (around **8%**) only increasing the AMR capacity.
In fact for all the pickers and AMRs configurations scenario where the picker capacity is 4, implementing Smart AMRs policy deteriorate the simulation time (roughly between **-1,3%** and **-2%**)



The following slides present an evaluation of the impact of storage, routing, and batch policies on productivity (measured as simulation time), considering the different configurations of pickers, AMRs, and AMR capacity.

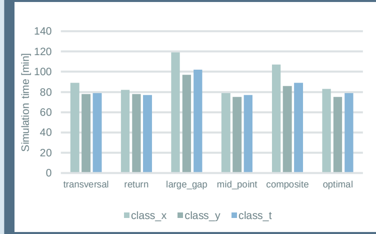
Productivity analysis: storage and routing policies

This part analyses the relationship between storage policy and routing policy in terms of productivity, considering variations in the number of pickers, AMRs, and AMR capacity.

First, it was determined that for each of the five cases considered, the optimal routing policy yields the best results for any warehouse configuration (an improvement with respect of random case ranging between **35.9%** and **39.7%** comparing different pickers and AMRs solution).

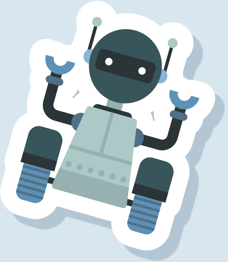
The best combination overall is the optimal routing policy combined with the class_y warehouse configuration (An improvement compared to the AS-IS case between **66.4%** and **78.9%**, considering the different combinations of pickers and AMRs.

2p2a4c



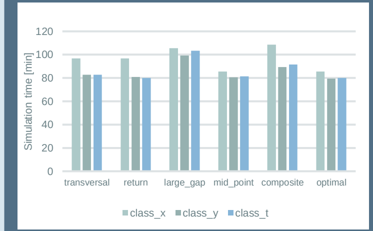
2p3a4c

2p2aXc



3p3a4c

3p3aXc



Productivity analysis: batch size and batch type

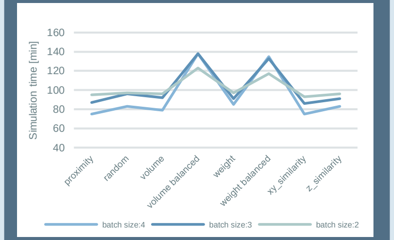
This section analyses the relationship between batch size and batch type in terms of productivity, considering variations in the number of pickers, AMRs, and AMR capacity.

XY similarity yields the best results compared to other batching policy types when varying batch size, the number of pickers, AMRs, and AMR capacity (around **13%** improvement in respect of random batching policy)
Weight_balanced and volume_balanced perform the worst (around **-45%** if compared with random batching policy) , which is understandable as they are designed to enhance worker satisfaction.

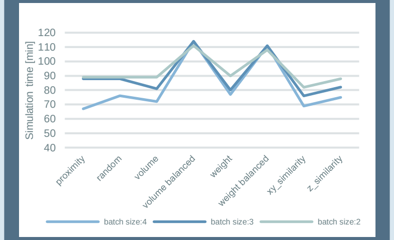
Analysing batch size, it is observed that increasing from 2 to 3 and from 3 to 4 consistently brings to a reduction of simulation time ranging between **16,1%** and **19,7%** except for Weight_balanced and Volume_balanced where the time performance deteriorate .

Nonetheless, the best results in each scenario are always achieved with the combination of batch size 4 and XY similarity.

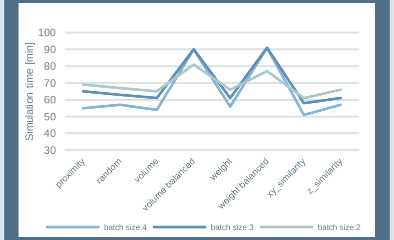
2p2a4c



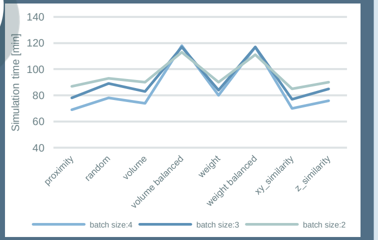
2p3a4c



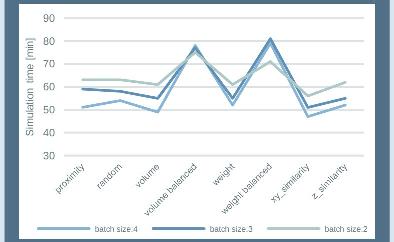
2p2aXc



3p3a4c

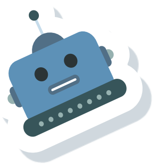


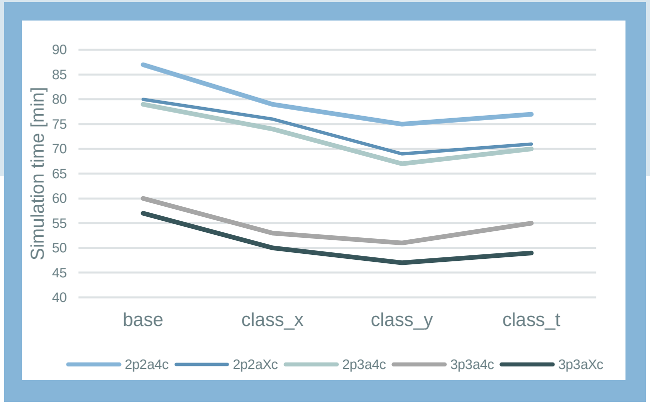
3p3aXc



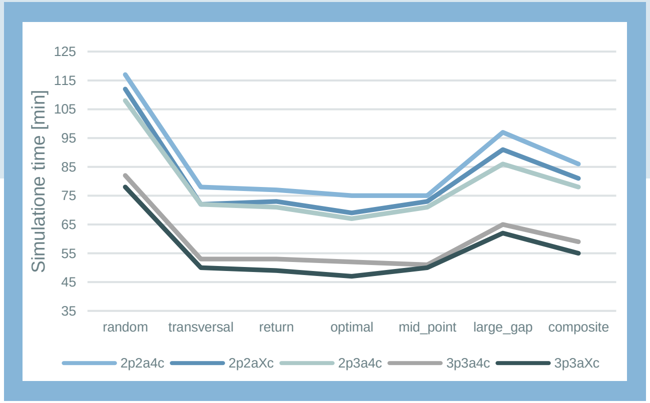
Productivity analysis: solution comparison

This section presents an analysis of the system considering variations in the number of AMRs (2 or 3), the number of pickers (always less than or equal to the number of robots), and the AMR capacity levels (4, 5, 6, 7, and 8).
The highest productivity is achieved with class storage, smart AMRs, and batching, confirming the previous slide. With a simulation time of 47 minutes, the optimal result uses Class_y storage, optimal routing, a batch size of 8, xy similarity batching, 3 pickers and 3 AMRs with a capacity of 8.

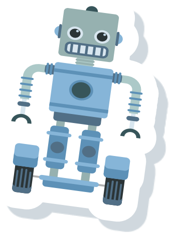




Storage policies



Routing policies



Observing both the graphs, it becomes evident that the addition of an extra AMR and an increase in its capacity positively impact simulation time, although the effect remains limited (ranging between **2.7%** and **11.3%**). Conversely, introducing an additional picker serves as a performance multiplier, leading to simulation time improvements of approximately **30%** when adding an AMR and around **35%** when both adding an AMR and increasing its capacity to 8.

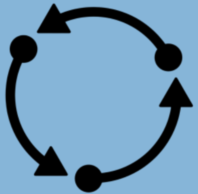
Productivity analysis: worker satisfaction analysis

To analyse worker satisfaction, four main drivers have been identified as potentially influencing the satisfaction of warehouse workers: the number of working hours spent in the warehouse (maybe considering the possibility of shorter shifts or reassignment to less physically demanding tasks in other sections), the number of picker steps, the total weight and volume managed during an order wave. Based on this considerations, two potential solutions have been developed.

Equitable workload

The total weight and volume of the order wave are evenly distributed among the various pickers. In particular, if the weight and volume carried by each picker fall within a ±2.5% range around the values of total weight/#pickers and total volume/#pickers, respectively.

Rotation of intensive workdays



The total weight and volume of the order wave are not evenly distributed among the pickers, who alternate between heavier and lighter workdays. It is assumed that there are no weight or volume limits on what employees can carry during a single simulation.

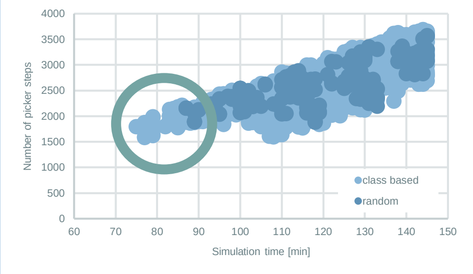
After analysing productivity, it was found that in the best cases, volumes and weights were not evenly distributed among the pickers. Therefore, the next slides will focus solely on the equitable workload solution, as the same considerations made during the productivity study can be also applied to the rotation of intensive workdays solution.

Equitable workload

*All graph on this page are drawn for the scenario with 2 pickers, 2 AMRs, and AMR capacity of 4. However, the graphs are similar also changing the number of pickers, number of AMRs, or AMR capacity except for Smart AMRs graph when capacity is bigger than 4. (see excel file)
** Only cases where the weight per picker is within 2,5% of the average weight per picker, and similarly for volume

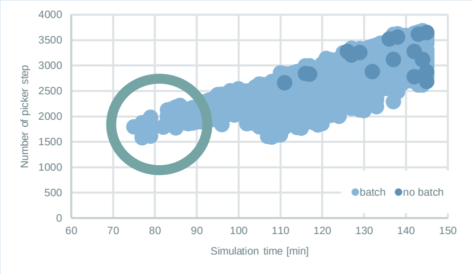
It is possible to separately observe the impact of introducing class-based, batch, and Smart AMRs policies. Specifically, the analysis includes both the baseline case with 2 pickers, 2 AMRs, and an AMR capacity of 4 (2p2A4c), as well as other notable configurations: 2p3a4c, 3p3a4c, 2p2aXc, and 3p3aXc.

Class based impact



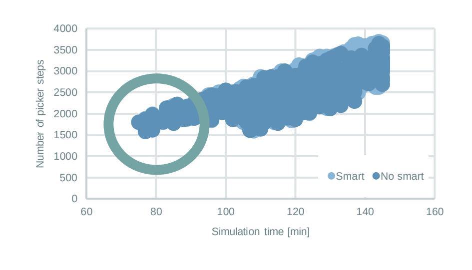
The introduction of class-based brings to a reduction of simulation time (and number of steps). Comparing the best case of class-based storage with the best case of random storage for all pickers and AMRs configurations there is always an improvement ranging between **12,5%** and **21,0%**

Batch impact



The introduction of batch results in a big reduction in simulation time (and number of steps). Comparing the best batch case with the best non-batch case leads to a reduction of simulation time ranging of **32,4%** and **37,9%** for all pickers and AMRs configurations. Furthermore, it is clear that the density of the number of balanced batch cases is much higher than the density of non-balanced batch cases

Smart AMR impact

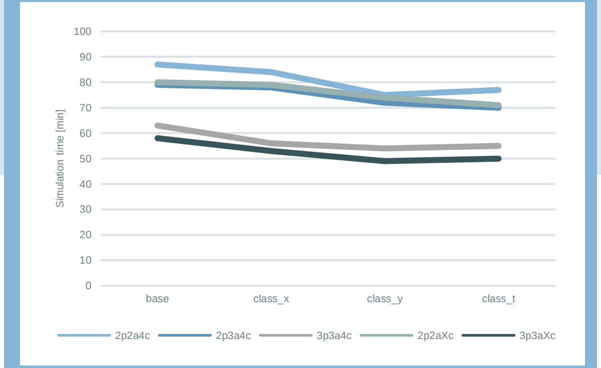


Introducing smart AMRs brings to a small reduction of simulation time (around **7%**) only increasing the AMR capacity.
In fact for all the pickers and AMRs configurations scenario where the picker capacity is 4, implementing Smart AMRs policy leads to negligible changes(between **-1,3%** and **+1,4%**)

The following slides present an evaluation of the impact of storage, routing, and batch policies on productivity (measured as simulation time), considering the different configurations of pickers, AMRs, and AMR capacity and the equitable workloads.

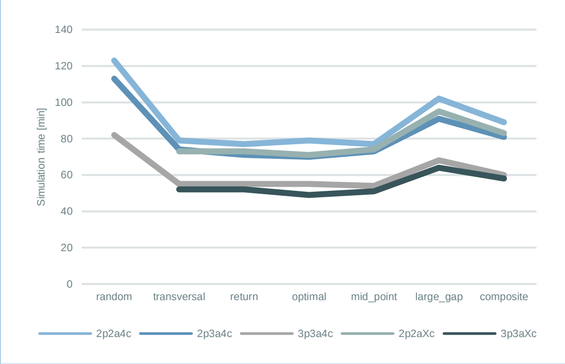
Equitable workload

This section presents an analysis of the equitable workloads solution considering variations in the number of AMRs (2 or 3), the number of pickers (always less than or equal to the number of robots), and the AMR capacity levels (4, 5, 6, 7, and 8).
The highest productivity is achieved with class storage and batching, confirming the previous slide. The optimal result, with a simulation time of 49 minutes, uses Class_y storage, optimal routing, a batch size of 4, volume batching type, 3 pickers and 3 AMRs.



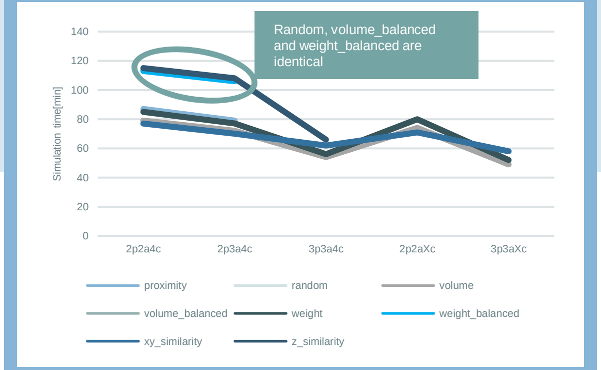
Storage policies

It is evident that adding a third AMR becomes beneficial when another picker is introduced (improvement of about **25%** and **30%** respectively for AMRs capacity of 4 and 8). Furthermore, Class_y and Class_t prove to be the best picking policies (always characterized by an improvement between **7,5%** and **15,5%** if compared with the AS-Is warehouse policy)



Routing policies

Introducing any routing policy results in a significant productivity improvement for all configurations. Optimal proves to be the best routing, in fact for each configuration of pickers, number, and capacity of AMRs, there is a gain of more than **30%** compared to the random policy. Furthermore, also in this case adding a third AMR becomes beneficial when another picker is introduced.



Batching policies

Considering the various configurations of pickers, AMRs, and capacities, not all batching policies can deliver an acceptable result in terms of volume and weight balance. The best result is achieved with a volume policy that Which always ensures a reduction in simulation time that is always around **30%** if compared with random batch type policy.

Productivity analysis: worker satisfaction analysis

In terms of worker satisfaction, after a thorough analysis of the two potential solutions (rotation of intensive workdays and equitable workloads), the following conclusions were reached:

Equitable workloads

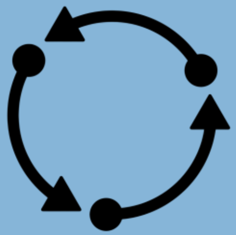
2p2a4c

Simulation time: 76 min
Average employe steps: 2039
Storage policy: Class t
Routing policy: Mid_point
Batching policy: XY similarity
Batch size: 4
Smart AMRs: indifferent

3p3aXc

Simulation time: 49 min
Average employe steps: 1495
Storage policy: Class y
Routing policy: Optimal
Batching policy: Volume
Batch size: 4
Smart AMRs: indifferent
AMR capacity: 8

Rotation of intensive workdays



2p2a4c

Simulation time: 76 min
Average employe steps: 2390
Storage policy: Class y
Routing policy: Mid_point
Batching policy: XY similarity
Batch size: 4
Smart AMRs: yes

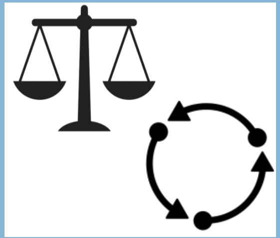
3p3aXc

Simulation time: 47 min
Average employe steps: 1315
Storage policy: Class y
Routing policy: Optimal
Batching policy: XY similarity
Batch size: 4
Smart AMRs: yes
AMR capacity: 8

Conclusion

Productivity

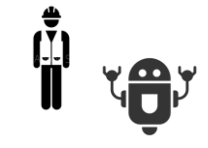
To optimise warehouse productivity, several strategies were implemented, reducing the simulation time from an initial 223 minutes to 75 minutes. The most effective configuration involved adopting a triangular class-based storage system, an optimal routing policy, a batch size of 2, XY similarity-based batch grouping, and using smart AMRs. These changes led to a significant improvement of **66.4%**, with relatively low costs primarily associated with inventory reorganisation and picker training.
Additionally, the impact of introducing an extra AMR and picker was assessed. This new configuration further reduced the simulation time to 47 minutes, achieving a **78.9%** improvement compared to the initial value. However, the additional investment may only be justified if there is an urgent need to enhance productivity. Specifically, the cost of acquiring an extra AMR and hiring a new staff member would result in only a modest reduction of 29 minutes to wave processing time, making the return on investment questionable.



Worker satisfaction

The analysis also considered worker satisfaction by developing a solution that enhances productivity while ensuring an equitable distribution of weight and volume among operators. Specifically, the solution aimed to maintain weight and volume within a 2.5% range around the average value per operator. The result was a **65.9%** increase in productivity, with an additional **77.5%** improvement when an extra AMR and a new warehouse worker were introduced. This solution would reduce the physical strain on operators by preventing excessively demanding workdays and allow for reassignment to less physically intensive tasks in other areas of the warehouse, facilitated by the increase in overall productivity. It is also important to highlight that the implementation of worker satisfaction results in a productivity decrease that is almost negligible (**<1.5%**), whether or not a new AMR and picker are introduced. Therefore, its implementation appears to be a choice worth seriously considering

Conclusion

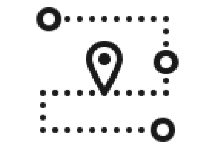


AMRs and pickers

When considering the AMRs, a key factor to address is their need for recharging. Given that the working time has been reduced to 49 minutes (or 76 minutes in the worst-case scenario), it can be reasonably assumed that the battery life of the AMRs could be designed to support these operating durations without issue. Regarding the potential acquisition of an additional AMR, this would only be justifiable if accompanied by the hiring of a new operator. This relationship becomes particularly evident when analysing the graphs on pages 13 and 16, which clearly demonstrate the need for increased workforce AMR capacity to fully optimize the benefits of an additional AMR in the system.

AMRs Capacity

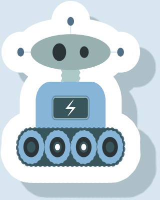
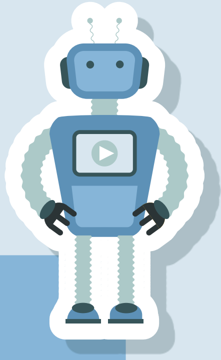
An observation that can be drawn from the graphs is that, as the global optimum is approached, the AMR capacity not have a significant impact on the final processing time. This is evident in the data: comparing the cases 2p2a4c and 3p3Xc respectively with the cases 2p2aXc and 3p3aXc there is only a a gain of only 6 and 4 minutes.
Consequently, it can be concluded that optimising capacity is not a critical factor, so AMRs capacity cannot be considered a worthwhile investment.



Optimal path

Another important consideration is that the optimal strategy may not align intuitively with the pickers' perspective, as it does not follow explicit logic from a human standpoint. For this reason, it may be beneficial to adopt routing strategies that are more straightforward to implement. In line with this reasoning the return strategy appears to be the most effective policy in terms of balancing worker satisfaction and productivity, in fact with all other factors being equal, the return policy deviates from the optimal by at most 4 minutes

References

Warehouse management references

Routing policies:

- ‘Logistics News: Designing the Most Effective Order Pick Routing in the DC’. Accessed 15 November 2024. https://www.scdigest.com/assets/experts/holste_09-07-02.php.

Storage,routing and batching policies:

- Ballou, Ronald H., and Ronald H. Ballou. *Business Logistics, Supply Chain Management: Planning, Organizing, and Controlling the Supply Chain*. 5. ed., Internat. ed. Pearson Education International. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.
- Chopra, Sunil. *Supply Chain Management: Strategy, Planning, and Operation*. Seventh edition, Global edition. Harlow, England London New York Boston San Francisco Toronto Sydney Dubai Singapore Hong Kong Tokyo Seoul Taipei New Delhi Cape Town Sao Paulo Mexico City Madrid Amsterdam Munich Paris Milan: Pearson, 2019.
- Coyle, John Joseph, ed. *Supply Chain Management: A Logistics Perspective*. 8. ed. Mason, Ohio: South-Western, 2009.

Code and simulation references

A* Algorithm:

- Halaoui, Hatem F. ‘Using Smart A* Algorithm to Solve TSP Navigation Problem’, 2018.

Travelling Salesman problem:

- Triantafyllou, Serafeim A., and Theodosios Sapounidis. ‘Algorithmic Approaches to the Traveling Salesman Problem for the Development of Students’ Computational Thinking’. In *Artificial Intelligence, Big Data, IoT and BlockChain in Healthcare: From Concepts to Applications*, edited by Yousef Farhaoui, 6:509–23. Information Systems Engineering and Management. Cham: Springer Nature Switzerland, 2024. https://doi.org/10.1007/978-3-031-65018-5_48.

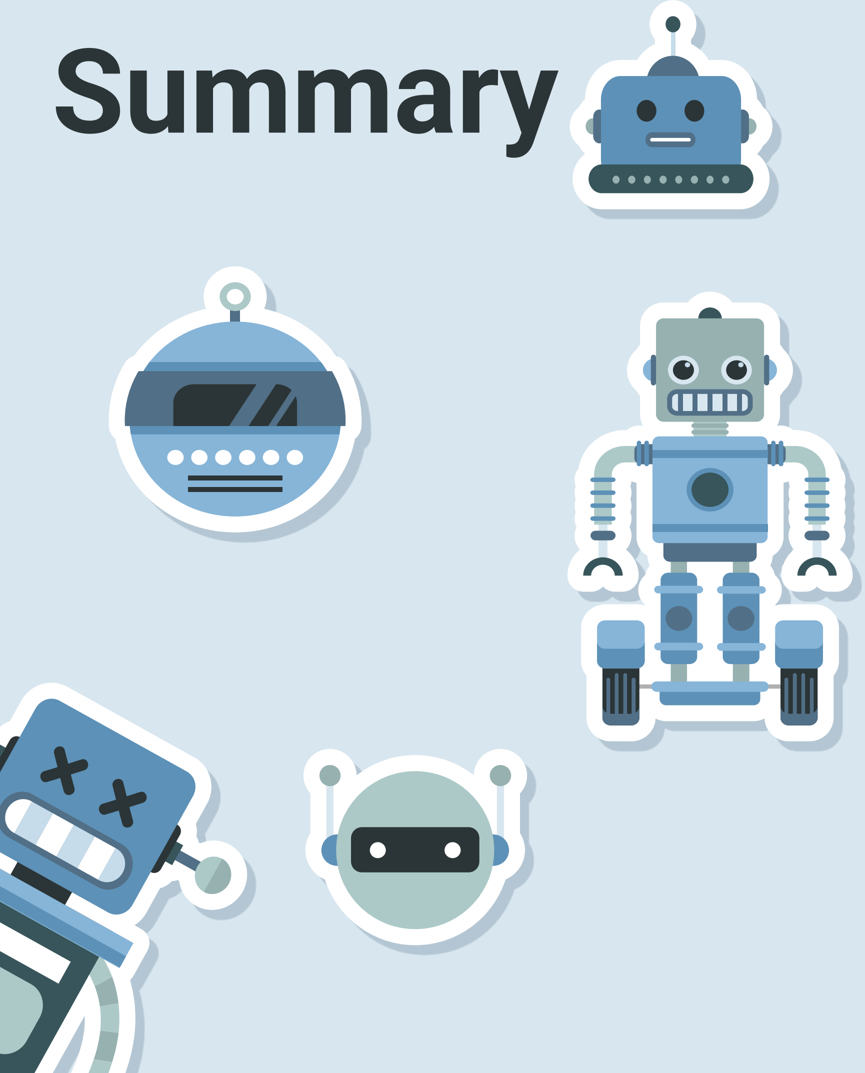
Jaccard distances and hierarchical clustering:

- Costa, Luciano da F. ‘Further Generalizations of the Jaccard Index’. arXiv, 18 November 2021. https://doi.org/10.48550/arXiv.2110.09619.
- Ran, Xingcheng, Yue Xi, Yonggang Lu, Xiangwen Wang, and Zhenyu Lu. ‘Comprehensive Survey on Hierarchical Clustering Algorithms and the Recent Developments’. *Artificial Intelligence Review* 56, no. 8 (August 2023): 8219–64. https://doi.org/10.1007/s10462-022-10366-3.

Assignment 2

Advanced modeling for operations
Professor: Elena Tappia
A.Y. 2024-2025

Summary



Objective

The focus on this second part is to assess the impact of the main picking policies on a particular KPI: **the orders per hour.** In particular, the objective for the system is to maximize the orders processed every hour.

In doing so, all possible factors will be analysed, both mere productivity-oriented policies (storage policies, routing policies and batching policies) and human-wellness centric ones (such as order batching in a way that the weight is balanced across each batch, or such that the picking height is quite constant), which were introduced during the first assignment and have shown good chances for improvements in the original system.

Hypothesis

Across all the possible systems analysed in part 1, all the dissertation of this second assignment will focus on the **system with 2 pickers and 2 AMRs, each with the maximum capacity of 4 orders**.

This is due to two main reasons:

- it is the original scenario
- it is the setting which generated the best performances during the phase one analysis, considering that the improvement achieved with smart AMRs, the additional capacity and the other required modifications led to marginal improvements in a cost-benefit perspective, which should be analysed more in depth but it is out of the project scope.

During the first assignment 5 main management choices were analysed

Storage Policies: random storage and Access Index based storage were analysed, with the latter in 3 possible configurations: triangular in respect to the I/O point, along x axis and along y axis, totalling 4 policies

Routing Policies: 6 main policies were identified and analysed: transversal, return, mid-point return policy, large-gap return policy, a composite policy and an optimal policy which decides what is the best path each time.

Single vs Batch Order Picking: here only two possibilities were considered: batching or not batching

Batch Sizing: 4 possible batch sizes were considered, each defined also on the batching policy, ranging from 1 order per each (equal to not-batching), up to 4.

Batching Policies: 8 batching policies were considered, accounting also for those policies that benefitted the health and well-being of the pickers

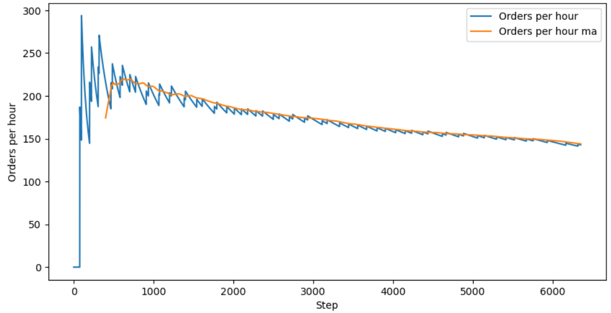
Among these, the most promising combinations were obtained by batching orders, whose size gave different performance levels based on the batching policies, but among all a batching of 4 orders with a proximity-based batching and an optimal routing policy resulted in the best performances. Unfortunately, all these results were drawn on a single and relatively small sample of orders (250). A further assessment of these performances is needed to understand both the quality and the stability of these solutions.

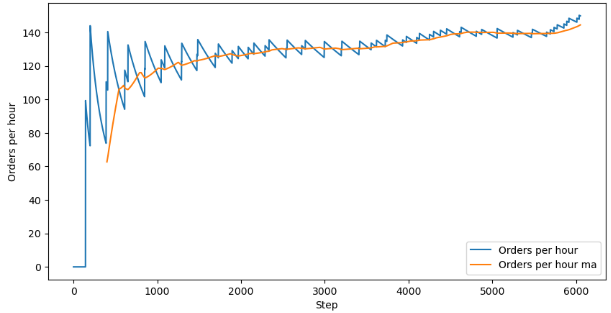
First Results & Data Transformation

While running the simulations with the different combinations of managerial policies, the patterns on the graphs besides emerged.

A warm-up period is clearly present in all the simulations, to then converge to a value around 150 orders/hours, for what concerns the KPI of interest. And two main ways to approach that value emerged, either asymptotically from above, or from below.

Due to the behaviour and the very low smoothness of the graphs, it was decided to proceed with a moving average. Having 6000 datapoints (ca.), the maximum width of the window should have been 1500, but after several tries, it was decided to use moving window of 400, allowing both to smooth out the results and to avoid too much information loss.





The upper couple of graphs represents the system performance with a class_y storage policy, an optimal routing policy, with order batching (batch size=4), and a proximity_xy policy, whereas the lower couple of graphs represents the same system performance with a volume-based batching policy

Design of Experiment

To assess the impact of each managerial policy, each possible combination of routing policy, batching policy and storage policy were analysed through a Design of Experiments approach. For each of these combinations the confidence intervals where analysed, helping to identify what are the critical factors, their interactions, and the optimal conditions for improving the picking efficiency trying to minimize its variability.

To perform this analysis a **Full Factorial DOE was required**. The parameters chosen are two levels for each comparison if they don't overlap. Every simulation was run until it reached the required parameters for accuracy or a maximum of 100 runs. *The safety parameters were not considered because they always result in the lowest productivity, same for the smart AMR that is always set on false.*
For each run a randomly generated set of 250 orders was used, to have comparable results across runs.

Number of Experiments

To compute the total number of experiments to perform, each possible combination should be evaluated, to do so:

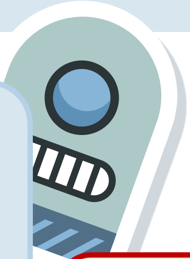
- If batching is performed, there could be 8 possible batching policies, and there are 4 possible storage policies considered, 6 routing policies are possible and 4 possible batch sizes, thus giving a total of
$$8 * 4 * 6 * 4 = 768\ experiments$$
- To those we shall sum those needed when batching is not performed, therefore, only the 4 storage policies and the 6 routing policies shall be considered, thus resulting in:
$$4 * 6 = 24\ experiments$$

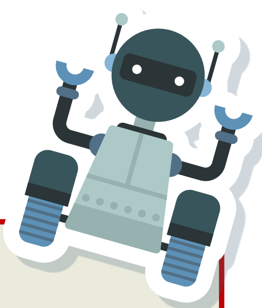
The total experiments to be performed are then the sum of these two values, adding up to ***792* experiments.**

DOE – Analysis Of Variance

As mentioned before, our objective is to conduct a DOE to understand which factors are the most relevant and give more variance for the orders/hours concerned. The factors initially considered factors were already mentioned, but for a better understanding of the problem, now will report all the levels for each factor:

- **Storage Policies:** ["class_y", "class_x", "class_t", "random"]
- **Routing Policies:** ["traversal", "mid_point", "large_gap", "return", "composite", "optimal"]
- **Batch:** [True, False]
- **Batching Policies:** ["proximity", "weight", "volume", "weight_balanced", "volume_balanced", "xy_similarity", "z_similarity", "random"]
- **Batch Sizes Policies:** [1, 2, 3, 4]



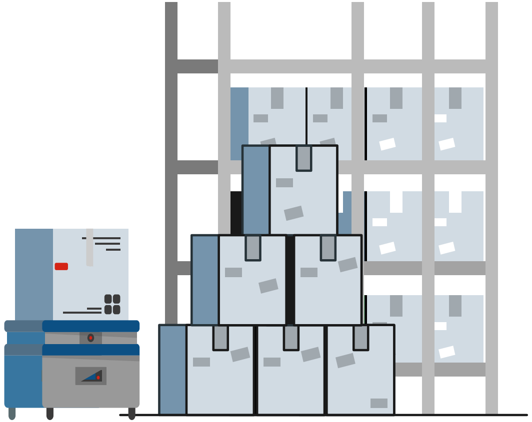


ATTENTION:
To reduce the number of factors and optimise the analysis, the Batch factor was embedded with Batching Policies, adding to this last one a new batching type called “None.” Moreover, the Batch Size Policies must be considered conditional factors because they are valid for certain factors. This final consideration means the **Desing Matrix** will not have a classical form.

DOE - ANOVA

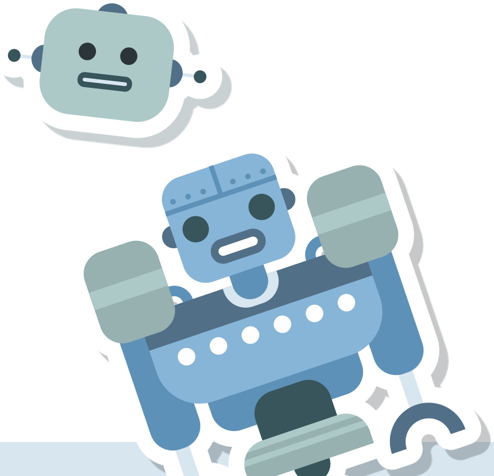
The first ANOVA was made with all the factors, even the **batch** (that was embedded in **batch_type**). The first result to notice, but not a surprise, is that batch is not significant for the variance of the process because it was embedded in the other factor, that how expected is very substantial how can be seen from the p-value. Then, an unexpected outcome is that storage is not statistically significant for the variance of the process. This is due to the small dimension of the warehouse, so the typology of storage is not very relevant. Finally, from this ANOVA, we can remove the factors that we consider insignificant, highlighted in red in the table.

	df	sum_sq	mean_sq	F	PR(>F)
C(storage)	3.0	8.077969	2.692656	0.019003	9.964379e-01
C(pick)	5.0	100697.473675	20139.494735	142.130875	7.585305e-107
C(batch)	1.0	4.140702	4.140702	0.029222	8.643116e-01
C(batch_type)	8.0	130284.791574	16285.598947	114.932696	5.105975e-126
C(batch_size)	3.0	244324.812786	81441.604262	574.759529	3.398118e-196
Residual	772.0	109389.954049	141.696832	NaN	NaN



DOE - ANOVA

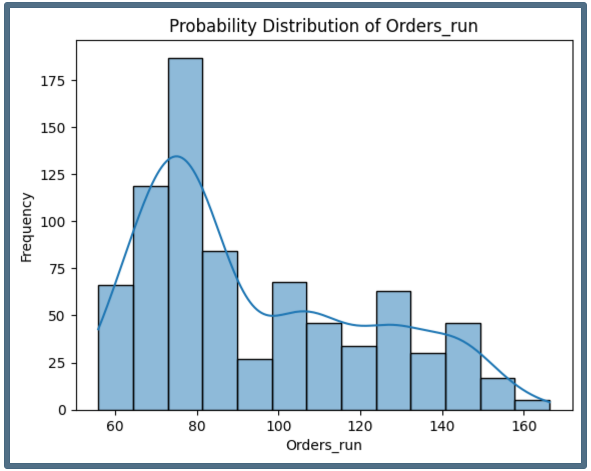
After the first ANOVA to identify the significant factors, another was performed to understand which interactions of the three remaining factors (**batch_type, batch_size, pick**). After different tests, even the three-way interaction (the maximum available) is essential. This indicates that there is a strong link between the three factors. Moreover, even if the line highlighted should be removed because it is not statistically significant, this wasn't done because in ANOVA, if an interaction between two factors is substantial, the main effects must be retained—even if not substantial—because the interaction depends on and cannot be interpreted without the context provided by the main effects.

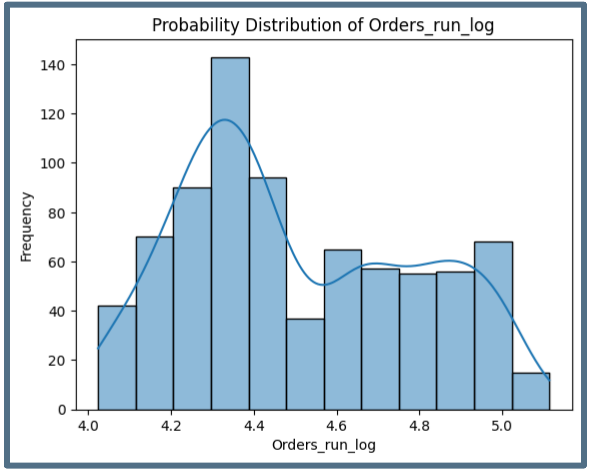


	df	sum_sq	mean_sq	F	PR(>F)
C(pick)	5	10.75571	2.151141	18479.24	0
C(batch_type)	8	15.81893	1.977366	16986.44	0
C(batch_size)	3	9.26E-13	3.09E-13	2.65E-09	1
C(pick):C(batch_type)	40	1.101726	0.027543	236.6076	0
C(pick):C(batch_size)	15	9.21E-13	6.14E-14	5.28E-10	1
C(batch_type):C(batch_size)	24	32.89887	1.370786	11775.65	0
C(pick):C(batch_type):C(batch_size)	120	2.064111	0.017201	147.7634	0
Residual	594	0.069147	0.000116	NaN	NaN

DOE – Data Transformation

LOG TRANSFORMATION:
The data were transformed using a logarithmic transformation because the ANOVA residuals did not meet the assumption of normality, as assessed through diagnostic tests. This transformation was applied to stabilise the variance and improve the symmetry of the residual distribution, ensuring the validity of the ANOVA results. After the transformation, the residuals were re-evaluated, confirming that they now approximated a normal distribution.





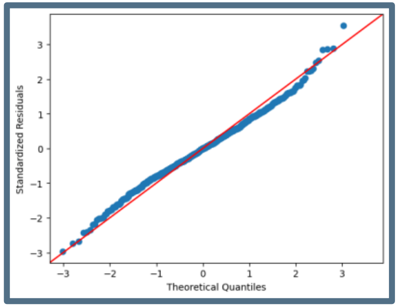
DOE – OLS and Hypothesis

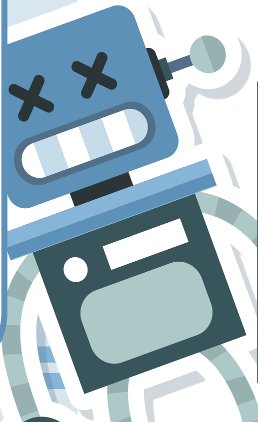
After finding a model with an ANOVO, there's the need to check if it respects the hypothesis and check with OLS the part of variance explained by the model:

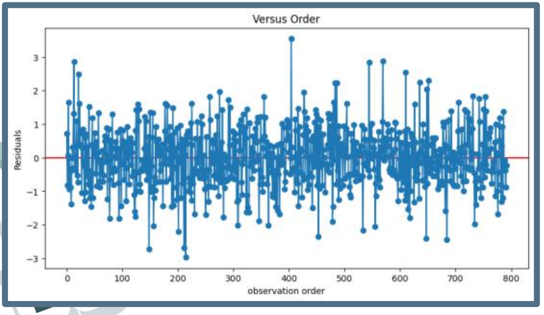
1. **Normality of Residuals:** The Anderson-Darling test confirms that the residuals follow a normal distribution. The Test statistic is 0.7026, which falls below the critical value of 0.783 at the 5% significance level. Therefore, the null hypothesis of normality cannot be rejected, indicating that the residuals are normally distributed.
2. **Independence of Residuals:** Visual inspection of the residuals versus order plot shows no discernible pattern, confirming the independence of residuals.
3. **Adjusted R-Squared:** The adjusted R2R 2 value of 0.999 indicates that the model explains almost all the variability of the dependent variable, with minimal overfitting, despite adjusting for the number of predictors. Moreover, the most influential predictor is the **pick**.

*More details about the OLS can be seen in the file attached to the presentation.

Metric	Value
Dependent Variable	Orders_run_log
R-squared	0.999
Adjusted R-squared	0.999
Model	OLS
Method	Least Squares
Log-Likelihood	2577.3
Number of Observations	792
AIC	-4759
BIC	-3833
Degrees of Freedom (Residuals)	594
Degrees of Freedom (Model)	197





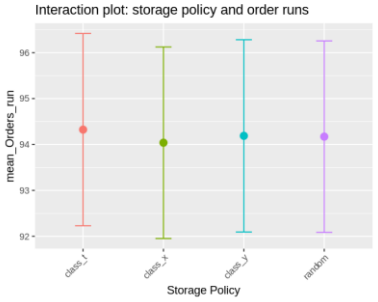


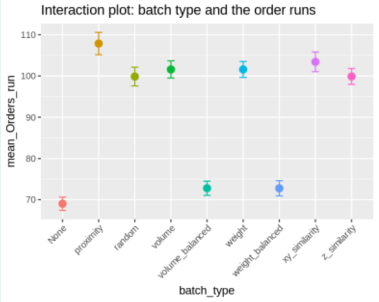
DOE – Factors Effects

Finally the *main effects* and *interactions* plot were produced here there are some example. All the graph are attached to the presentation this are only some preliminar insight. In fact in next slides the will an analysis of the result and more interestant consideration comeout from the DOE.

Interactions between policies and target variable

To ease the analysis, where batching was not applied, the batch size was set to 1 and the information was not lost, but included in the batching policies as "None" category. In the following plots the interactions between the managerial policies and the target variable (Orders/hour) are shown.
For each managerial policy tested, the average value of the order/hour was computed, together with its confidence interval at 95% confidence level, obtained by averaging all the intervals obtained for each policy.





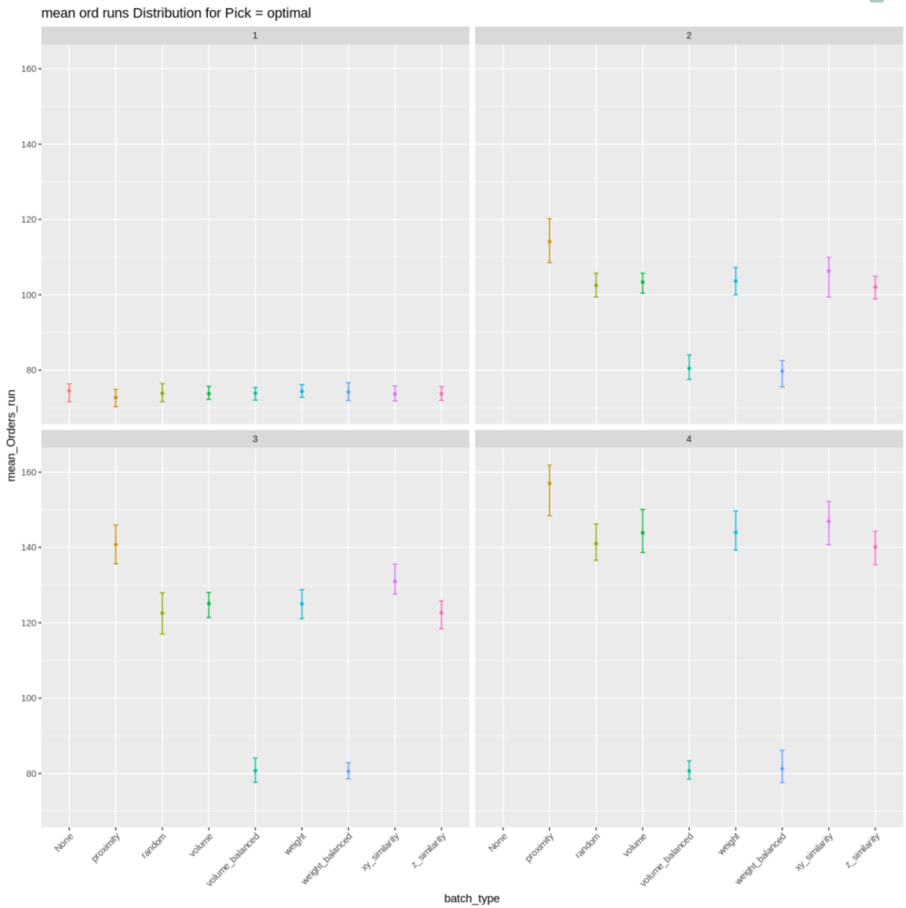
Interesting Outcomes

- **Random vs Class Based Storage:** by comparing the different storage policies both in terms productivity-based and confidence interval they do not seem to have significantly different performances. Moreover, the random storage policy seems to have performances closer to a class_y storage policy. This could be traced back to the relatively small size of the warehouse, which is only 500 m^2, therefore the way the storage is allocated is less influential than it would be in a bigger space.
- **Single Order vs Batch Picking:** not batching the orders leads to the worst system performances, as the third plot shows, while batching orders can increase the system productivity up to almost 70%, moving from an average of 69 orders/hour up to around 116 when using a batch size of 4!
- **Productivity- vs Human-Dirven Order Assignment:** remembering that the human-driven policies are embedded into the batching policies as batch types for volume, weight and z-similarity, the last plot shows that those policies driven by productivity lead to higher results for what concerns the orders/hour KPI, which could be expected since it is a productivity based variable, nonetheless the human-driven KPIs show promising results, not falling too far behind the aforementioned ones. A particular note should be made on the volume and weight balanced policies, which show up among the worst performers, together with the "None" class, yielding a performance which is on average 30% worse than the other policies.

Additional Outcomes: routing policies seem to have a sensible impact on the metric of interest, with the transversal policy appearing to be least promising one, together with the large gap routing. On the other hand, the mid-point return, the pure return policy and the optimal policy appear to be the best performers, with 25% more orders processed each hour (ca.) compared to the previous two policies. It is also worth to mention that the policy which composes transversal, and routing policy delivers a performance which is midway between the pure policies!
A further analysis on the cross influence of factors will be carried out in the upcoming slides, highlighting for the top 3 routing policy how both the batch size and the batch type affects the system performance. Moreover, this will be a precautionary analysis: for each combination of policies, the maximum of the upper bounds and the minimum of the lower bounds of the confidence intervals will be considered, showing us the highest variance that the process could assume, and providing valuable information for a more comprehensive analysis.

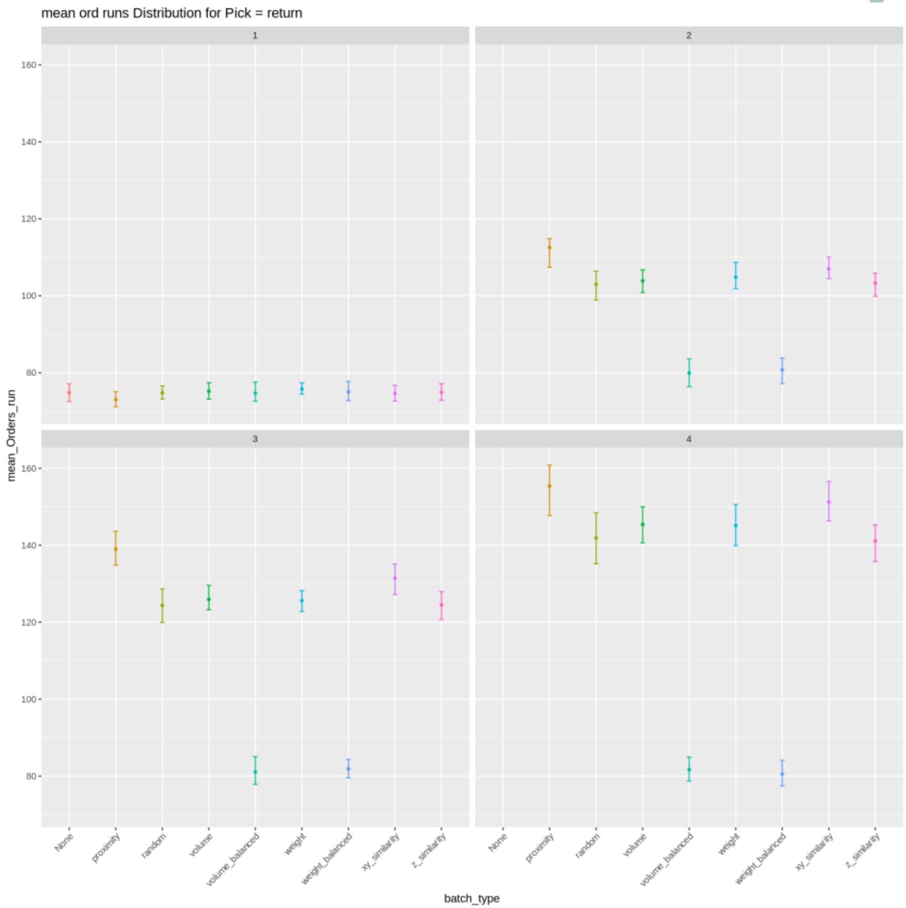
Optimal

In the previous analysis, the one deemed as Optimal routing policy was among the three most promising ones. By delving into the cross interactions among the policies, it is possible to see that the combination of the three policies together enhances the system performances!
Across all combinations, the one that allows to achieve the highest performances with the optimal routing policy is with the maximum batch size, and a batching policy by proximity. An interesting observation, is that with this combination of policies, the error bars get some overlapping with the xy_similarity, the volume and the weight based batching policies, which also shows a smaller error bar. This means that the despite the very good the performances, the proximity based batching policy shows also the highest variability, which could be a downside when looking at the performances consistency across runs.
On the other hand, by looking at the other batching sizes, the overlapping effect is small and almost negligible, while with size one all the polices overlap making it very difficult to identify a true “best policy”.



Return

The Return routing policy yields very similar results to the Optimal one, reaching the best performances with the highest batch size, and the Proximity-based batching policy, where its error bars just reach above 160 orders/hour.
While the two worse policies seem to remain the weight and the volume balanced ones, something interesting happens to the xy_similarity policy. Focusing on the batch size 4, the aforementioned policy yields better performances than with the optimal routing policy, getting closer to the proximity-based batching, thus increasing the overlapping effect observed before. Instead, both the volume and weight based policies remained quite stable, and similarly the z_similarity policy.
Another interesting observation is that the error bar for the proximity policy reduced with the return policy compared with the optimal one and similarly for the random storage, by looking at batch sizes 2 and 3 as well, while the volume, the weight and the z_simlarty policies show quite stable error bars leading to consistent performances across routing policies.

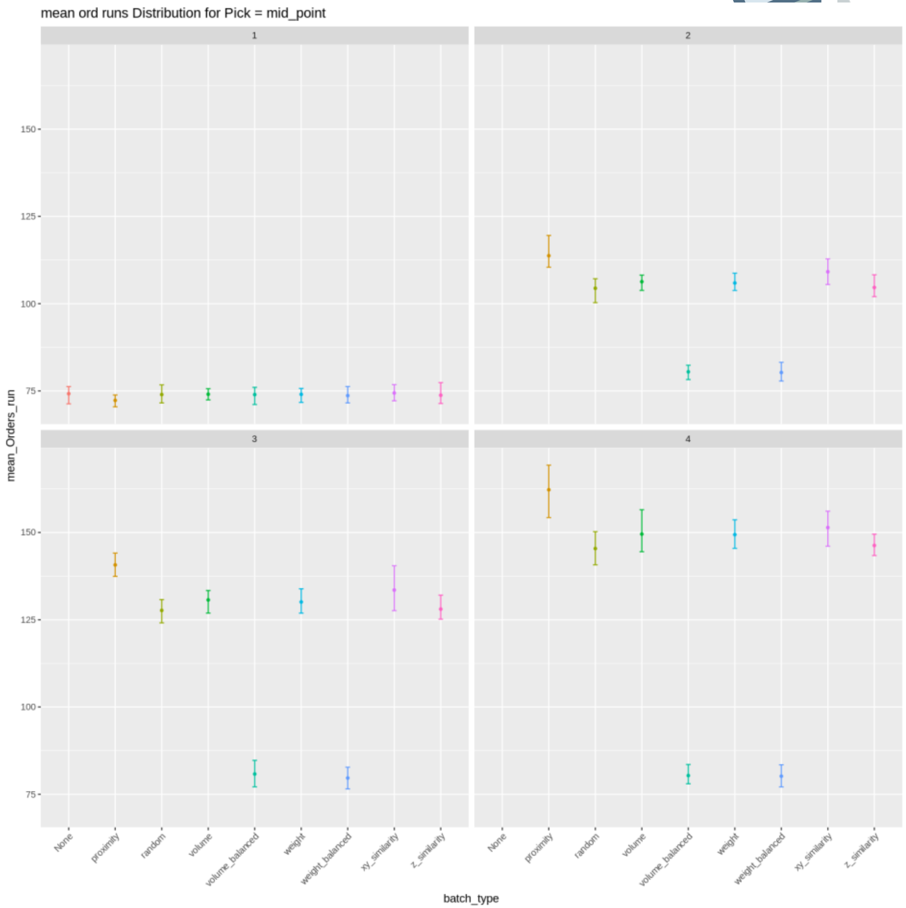


Mid-point Return

Lastly, by looking at the Mid-Point Return and being aware of the different scale of the graphs, it can be highlighted that this policy leads to the best results across all the ones analysed, averaging almost 162.5 orders/hour, and varying up to almost 170.

With this routing policy the xy_similarity batching yields performances that are very similar to the ones of both the weight and the volume based batching, where the former shows quite a reduced error bar.

While the proximity variability seems to be quite stable, the other policies seem to have a smaller error bar (still not too different from the previous two policies, which should lead to more stable performances between each run.



Conclusions

Summing up all the findings, it is possible to say that:

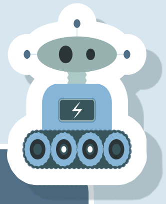
Key Result: all the performed analysis demonstrated how managerial policies interact with each other and influence the target KPI. Moreover, they pointed out how the proximity-based batching, with batch size of 4 and mid-point return routing yields the best results, on average, balancing both high performances and system adaptability, highlighting the susceptibility of this policy to variability though.

Additional Findings

- **Marginal Storage Impact:** Storage policies played a minimal role due to warehouse constraints. This is very important because the storage policies are the hardest to change since among all the considered policies are the ones requiring a physical change in the picking area set-up. This leads to two main considerations: firstly, if the warehouse setup remains unchanged, the picking system can be quite flexible among the other policies, and switching between them could be almost inexpensive, allowing the system to adapt to different workload if it faces seasonality; on the other hand, if in the future this space is set to be expanded the effects of the storage policies could play a decisive role.
- **Batching Synergy**: Batch size of 4 showed the highest efficiency, but integrating batching with routing policies (e.g., Mid-Point Return) further reduced variability and boosted throughput.
- **Routing Impact**: While Mid-Point Return routing was superior, the Optimal policy emerged as a viable alternative even though with higher variability in some scenarios, but still ensuring consistent performance.

Future Opportunities: Explore cost-effective ways to reduce variability in top-performing systems

References



- **Anderson, T.W., Darling, D.A., 1954. A Test of Goodness of Fit**. Journal of the American Statistical Association 49, 765–769. https://doi.org/10.1080/01621459.1954.10501232
- **Box, G.E.P., Cox, D.R., 1964. An Analysis of Transformations**. Journal of the Royal Statistical Society Series B: Statistical Methodology 26, 211–243. https://doi.org/10.1111/j.2517-6161.1964.tb00553.x
- **Gruber, M.H.J., 2013. Matrix algebra for linear models**. John Wiley & Sons, Inc, Hoboken, New Jersey.
- **Kutner, M.H., Nachtsheim, C.J., Neter, J., Li, W., Nachtsheim, C. (Eds.), 2005. Applied linear statistical models**, 5. ed., internat. ed. ed, McGraw-Hill/Irwin series Operations and decision sciences. McGraw-Hill, Boston, Mass.
- **Montgomery, D.C., 2017. Design and analysis of experiments**, Ninth edition. ed. John Wiley & Sons, Inc, Hoboken, NJ.
- **Shapiro, S.S., Wilk, M.B., 1965. An analysis of variance test for normality** (complete samples). Biometrika 52, 591–611. https://doi.org/10.1093/biomet/52.3-4.591
- **Tukey, J.W., 1977. Exploratory data analysis**, Addison-Wesley series in behavioral science. Addison-Wesley Pub. Co, Reading, Mass.