Journal of Industrial Information Integration 40 (2024) 100634

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.sciencedirect.com/journal/journal-of-industrial-information-integration

Exploring the relationship between routing policies and market demand heterogeneity: A simulation analysis with different hardware and software configurations in traditional warehouses

Michele Bocelli *, Eleonora Bottani, Andrea Volpi, Federico Solari, Natalya Lysova, Roberto Montanari

Department of Engineering for Industrial Systems and Technologies, University of Parma, Parco Area delle Scienze 181/A, Parma (PR), 43124, Italy

A R T I C L E I N F O

Keywords:
Warehouse
Routing policy
Input-output configuration
Simulation approach
Picking distance
Demand heterogeneity

A B S T R A C T

This study presents a simulative tool designed to evaluate quantitatively the performance of the order picking process in a traditional warehouse with double-sided racks; performance is measured as the distance covered by the picker when carrying out a single task. Using the simulation tool, different warehouse configurations can be generated by changing both hardware (shape factor and input-output position of the picker) and software (routing policy, distribution of the demand of products and number of items in the picking list) input parameters. Then, the outcomes (distance travelled) obtained with the tool are analysed, with a particular focus on the relationship between the routing policy and the level of heterogeneity in the market demand for the products available in the warehouse, which implies changes in the product allocation. The study concludes with important results about the impact of the warehouse geometry itself on the performance of the routing policy. Finally, suggestions are offered for future studies in the field.

1. Introduction and literature review

Order picking is a critical process in the supply chain, and any delays or errors in this process can have significant consequences on the entire supply chain [1]. The cost of order picking is also significant, especially for a manual process, which requires considerable amounts of resources and can contribute to more than 55 % of the total warehouse cost [2]. Most of the cost is due to the time spent by the pickers walking through the aisles, searching and picking items, and finally bringing them to a depot for consolidation [3].

A key issue in manual picking, therefore, is to make the process efficient by decreasing the “travel time” of pickers, which is a function of the travel distance to be covered [4]; hence, minimising this distance has been suggested across the years as the main leverage for optimizing the total picking time [5–9].

Solutions for minimising the travel distance range from structural (“hardware”) aspects to operational (“software”) ones [10]. Hardware aspects reflect unchangeable features of the warehouse, such as its size, layout or shape factor [11,12]. As opposed to these aspects, software elements refer to the operating conditions of the system, e.g., the use of high-level vs. low-level picking [13], the picking strategy [14–17]; the storage assignment policy [18,19].

These aspects all play an important role in the optimization of the picking process, both as single factors and in combination. However, for a long time, literature has focused on the analysis of one aspect, keeping the remaining factors unchanged. This choice obviously makes the optimization easier but does not allow for capturing possible correlations or combined effects of factors; this is why authors have started evaluating more design factors at a time [20].

According to both Manzini et al. [21] and [22], routing strategies and storage allocation policies (or routing coupled with batch picking) are among the factors that have been evaluated in conjunction, because of the relationship existing between storage and routing planning problems. Indeed, the length of a picking tour is expected to decrease when applying an allocation strategy different from the random storage, or when coupling a proper storage assignment with a suitable routing policy; in line with this, the recent review by Casella et al. [20] reports that the interest towards storage assignment logics has increased in time.

* Corresponding author.
E-mail addresses: michele.bocelli@unipr.it (M. Bocelli), eleonora.bottani@unipr.it (E. Bottani), andrea.volpi@unipr.it (A. Volpi), federico.solari@unipr.it (F. Solari), natalya.lysova@unipr.it (N. Lysova), roberto.montanari@unipr.it (R. Montanari).

https://doi.org/10.1016/j.jii.2024.100634

Available online 14 May 2024

Table 1
Nomenclature of the model evaluated.

Symbol	Description	Unit
PD	Picking Distance	m
APD	Average Picking Distance	m
M	Max *APD*	m
m	Min *APD*	m
Δ	Difference in *APD* with different routings	%
RP	Routing Policy	–
RS	Return Simple	–
RAD	Return Advanced	–
SSS	S-Shaped Simple	–
SSAD	S-Shaped Advanced	–
x_f	Warehouse Shape Factor	–
x_{fT}	Warehouse Shape Factor Target	–
x_{fR}	Warehouse Shape Factor Real	–
LA	Number of Longitudinal Aisles	–
CA	Number of Cross Aisles	–
a	Width of the warehouse	m
b	Depth of the warehouse	m
α	Width of the storage location	m
β	Depth of the storage location	m
w	Width of the aisles	m
I/O	Input-Output position of picker	–
SCP	Single Central Picking	–
SLP	Single Lateral Picking	–
OLPSS	Opposite Lateral Picking, Same Side	–
OCP	Opposite Central Picking	–
OSP	Opposite Side Picking	–
λ	Parameter of the demand distribution	–
i	i^{th} item in the warehouse	–
P	P^{th} storage location in the warehouse	–
n	number of items required to satisfy 80 % of market demand	–
N	Warehouse storage capacity	–
R(P)	Distance taken by the picker to reach the storage location (Ranking)	m
PL	Number of Items Picked up each mission	–
SMDP	Simulation Model Development Process	–
PDF	Probability Density Function	%
CDF	Cumulative Density Function	%
CF	Number of configurations	–
σ_{APD}	Standard deviation of APD	m
CL	Confidence level	%
CI	Confidence interval	m
t	Critical value associated with the t-Student distribution	–
repl	Numbers of replicates	–
k	k^{th} replicate	–

Besides the random allocation, used, e.g., by Roodbergen et al. [23], typical assignment criteria include turnover-based storage classes [24], popularity of the item over its life cycle [19] or density-turnover index [25]. Studies that evaluated the allocation logic coupled with the routing optimization have in general assumed class-based storage ([10], Table 1).

The allocation logic and the routing strategies have also been studied in conjunction with some warehouse characteristics, such as the layout (traditional or unconventional), the location of the input/output (*I/O*) depot, the shape factor, or the presence and number of cross aisles [20]. One of the first studies that evaluated more design factors has been by Petersen [26]. The author has analysed jointly six routing policies (including *Return* and *S-shaped*), four warehouse shape factors, two positions of the *I/O* depot and five lengths of the picking list, for a total of 240 scenarios, and determined through ANOVA, the statistical effects of the single factors, as well as of two-factor interactions, which were all found to be significant at p <-0.05. Specific outcomes indicate that the *Return* policy is generally ineffective compared to the *S-shaped* and that the central location of the *I/O* depot reduces the tour length. The analysis assumes random storage of items in the warehouse. The same author [7] has evaluated and compared various routing strategies coupled with random and volume-based storage. In this study, the warehouse layout (number of corridors, *I/O* depot and shape factor) is fixed; four routing policies (including *S-shaped*) are evaluated, as well as two variants of the volume-based storage (within the aisle and diagonal). Outcomes suggest that, in general, the *S-shaped* is not effective with short picking lists, and that compared to random storage, volume-based storage typically results in shorter distances, especially if used in conjunction with specific routing strategies. Later, Petersen and Aase [8] have developed a simulation model intended to investigate the effect of three picking logics, three storage policies (random, class-based and volume-based) and three routing strategies (including the *S-shaped*) on order picker travel distance in a fixed warehouse configuration. They found that both volume-based and class-based storage policies involve significantly less travel time than random storage; as far as the routing, switching from the *S-shaped* to the optimal routing offers only a moderate decrease in the travel distance. Dukic and Oluic [27] have evaluated the performances of routing, storage and order-batching methods in combination, as a function of the specific context (layout, order size and order picker capacity). They found that an appropriate combination of order picking methods has the potential to reduce the travel distance by up to 80 %.

The correlation between the number of aisles, picking list size and path length in a class-based storage environment has instead been examined by Rao and Adil [28] for a two-block warehouse, in which a *Return* routing policy is assumed. The goal of the study was to optimise the number of classes to the allocation and the number of aisles, to maximise the savings in travel distance compared to the random storage. The outcomes have shown that a number of classes between 2 and 3 results in a good saving in the travel distance. Van Gils et al. [24] have made the first attempt to analyse the relationships between storage allocation decisions, batching, zoning, and routing; the statistical significance of the outcomes was tested by a full factorial ANOVA. They have found that warehouses can significantly improve their performance if considering storage, batching, zone picking, and routing decisions simultaneously. In a subsequent publication, van Gils et al. [13] have embodied safety constraints, picker blocking, and high-level storage locations in the design of the order picking process, again modelled in terms of zoning, storage, batching, and routing decisions.

The outcomes above indicate that some studies have evaluated the impact of a combination of two/three factors on the order picking performance, but that, at the same time, some gaps still exist. Indeed, storage assignment and routing appear as the pair of factors most frequently evaluated in conjunction, but they are both operational (software) aspects of the picking process, according to the previous definition; relationships with the remaining factors, and especially with hardware aspects (e.g., the warehouse layout), as well as more complex intersections among picking design factors, remain less explored [29]. Consequently, the available literature still does not fully capture the interrelationships between the various design decisions of the picking process. This is the first research gap to which this paper aims to contribute.

A second consideration from the review above is that, for sure, when approaching the analysis of more picking design decisions simultaneously, the scenario becomes too complex to be solved using analytic methods [24], and thus simulation is typically used to reproduce and study the problem [30]. Decision support tools, based on simulation models, have indeed been proposed to the picking context, although by a limited number of authors only. Decision support tools for warehouse/material handling design have been developed by Manzini et al. [31] and Accorsi et al. [32]; despite the general scope, these tools also address some challenges relating to the order picking process, by offering insights on warehouse layout, product allocation, and routing. Bottani et al. [29] have developed a tool for fully modelling the picking process, taking into account the warehouse layout (shape factor and size) and the routing policy, while neglecting storage allocation logics (thus implicitly assuming random storage). Ozden et al. [33] have presented an open-source software that starts with a picking list, sets some parameters (e.g., cross aisle width), searches for the optimization of some warehouse features (e.g., shape factor), and finally applies a meta-heuristic algorithm to generate a wide set of warehouse designs

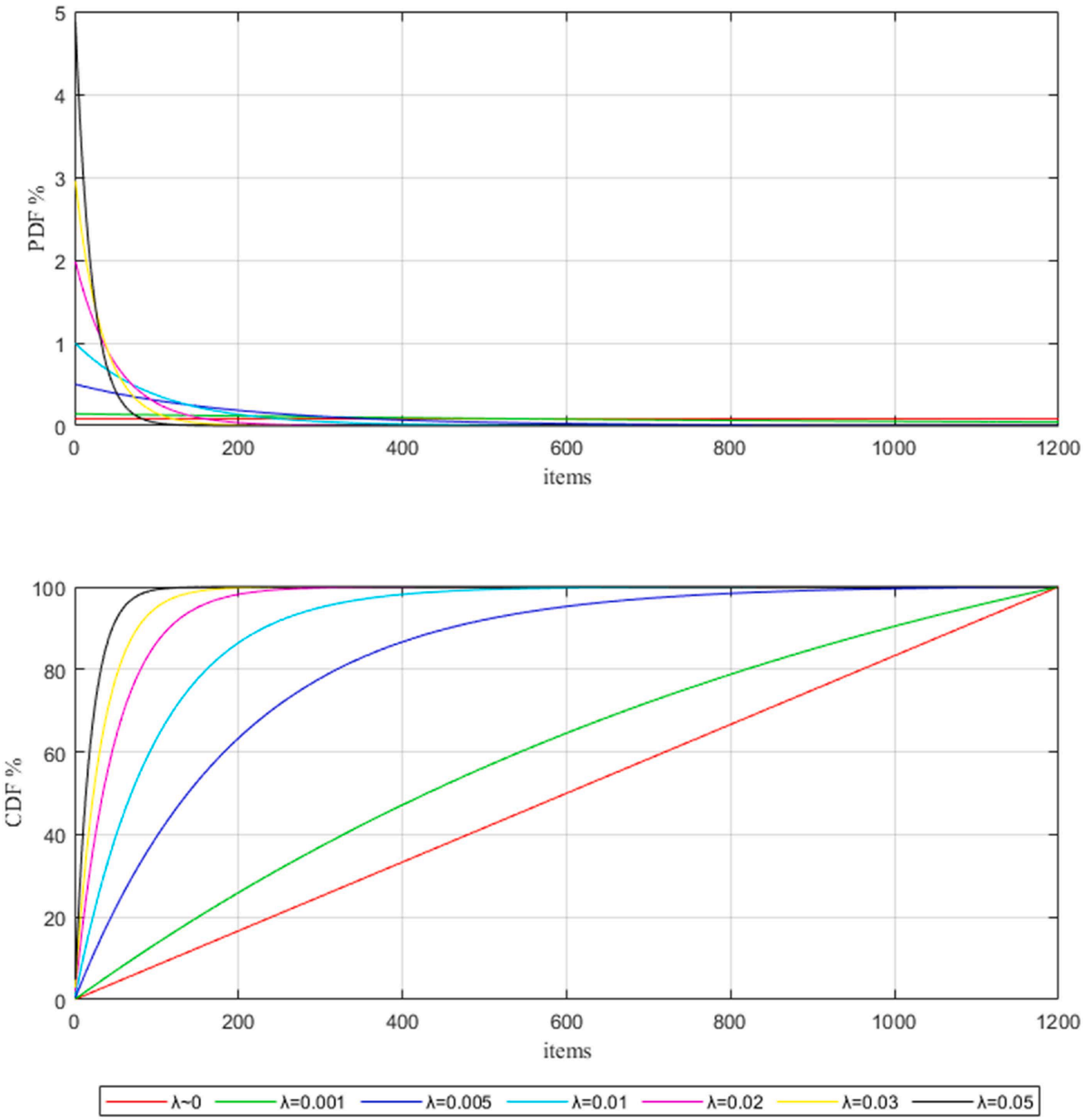

Fig. 1. Trends of PDF and CDF for different λ values and $N = 1200$.

Table 2
Items required to achieve 80 % probability of meeting market demands (*CDF*), with $N = 1200$.

λ	*n*	*n%*
~ 0	960	80 %
0.001	819	68 %
0.005	320	27 %
0.01	161	13 %
0.02	81	7 %
0.03	54	5 %
0.05	32	3 %

solutions, as well as to select the one which is expected to minimize the travel distance. Turnover-based storage is assumed as the allocation policy. Other tools for the design and analysis of picking systems in rectangular warehouses available online include the *interactive warehouse* and the *warehouse optimizer* [34,35]. These tools allow the layout of a regular warehouse to be reproduced by setting the number of aisles, blocks and locations per aisle side together with the depot location. Then, the user can generate a random picking list (of a given size) or directly select a set of picking locations on the warehouse layout. The tools include some known routing policies, while the allocation strategy is not taken into account; some limits also exist about the size of the warehouse that can be reproduced and optimized using these tools. Overall, the available tools for picking design are few and have some limitations; in this respect, a more general tool, able to capture additional design factors, could be useful to warehouse managers to evaluate the performance of the picking process as a function of the system settings. This paper will attempt to contribute to this second point.

Overall, this study is expected to enrich the available literature in

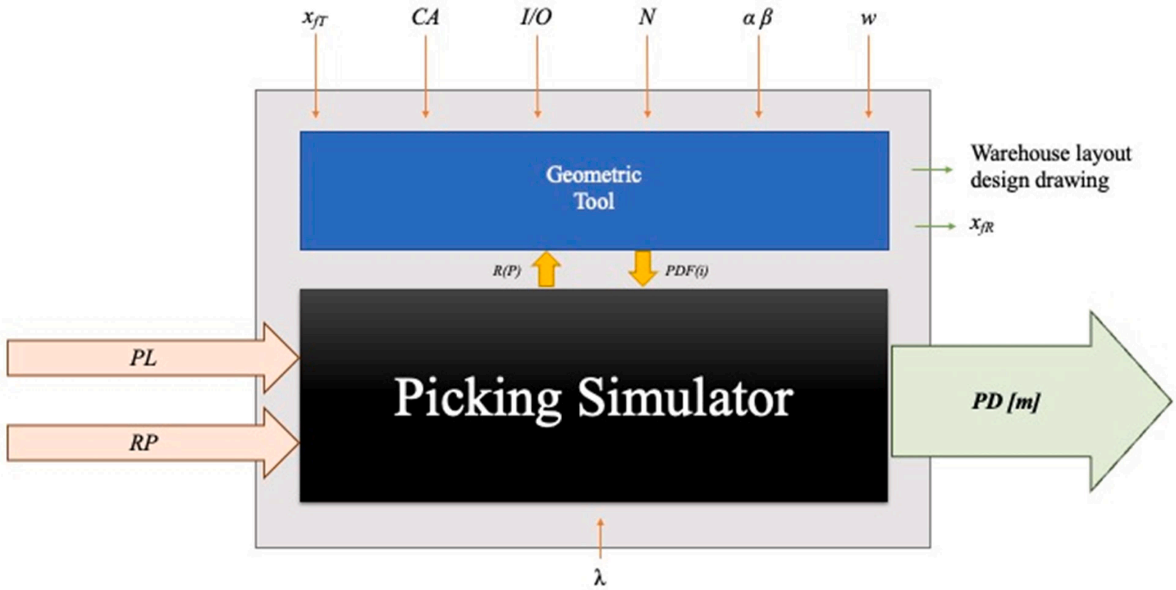

Fig. 2. Scheme of the simulation model.

Table 3
Shape factors considered in the study.

x_{fR}	*LA*	*a* [m]	*b* [m]
0.14	4	22	162
0.21	5	27.5	132
0.29	6	33	112
0.51	8	44	87
0.76	10	55	72
1.06	12	66	62
1.59	15	82.5	52
2.62	20	110	42
3.57	24	132	37
3.82	25	137.5	36

two interrelated ways. First, it will evaluate the performance of the order picking process as a function of several types of input, both software and hardware. Primary, the results will refer to the application of different routing policies in a context in which the characteristics of the “market demand” of the products handled in the warehouse can vary, thus implying changes to the product allocation. By “market demand” it is meant the fact that the items can have different requests: generally, there can be items more frequently and less frequently requested by the customers. Such modelling allows for capturing various scenarios, ranging from random storage (in case all items have the same behaviour with respect to the market request, which means that the allocation becomes indifferent), to dedicated assignments (in case the behaviour of items is different, and therefore, allocation decisions become relevant), and thus evaluating the effects of other design factors under a wide set of configurations. Introducing this element in the analysis allows for enriching the available literature by evaluating the performance of a warehouse subject to an “external” disturbance factor. Among the hardware aspects, the effect of the *I/O* depot will also be explored, because of the obvious relationships with both the allocation and routing logics. The second contribution comes from the approach followed for achieving the primary aim of the paper, and in particular, from the development of a simulation-based design tool for the picking process. The simulation model takes into account almost all picking design factors, and in particular, allocates the items in the warehouse as a function of the *I/O* depot, the shape factor and, most importantly, the characteristics of the market demand, whose analytic expression is embodied in the simulator. The analytic formulation of the market demand allows for the simulation of a wide range of scenarios that may be observed in applications.

The remainder of the paper is organised as follows. The next section details the materials and methods, including the nomenclature used in this study, the set of input and output factors taken into account in the analysis, and the development of the simulation model. Then, in Section 3 the model is used for reproducing a selected set of scenarios, resulting from the combination of some design factors, and related results are presented. Discussion and conclusions end the paper, by summarising the key outcomes, showing the advancement compared to the literature and suggesting future research directions.

2. Materials and methods

2.1. Nomenclature and acronyms

The nomenclature and acronyms used in this paper are presented in Table 1.

2.2. Problem definition

The simulation model was designed according to the Simulation Model Development Process (*SMDP*) approach proposed by Manuj et al. [30].

Minimisation of the *APD* through the appropriate tuning of the system parameters was identified as the objective of this study, because it can significantly reduce the time required to complete a picking task and, consequently, enhance the efficiency and productivity of the entire warehouse management process.

However, the analysis did not focus on *APD* minimisation only. Indeed, an additional critical analysis was carried out to highlight that, in some contexts, it is preferable to choose a configuration that is not fully optimized in terms of *APD*, but can generate operational benefits, e.g. simplification of the picking routes and/or the warehouse layout itself.

2.3. Independent variables

For developing the simulation model, which builds upon the previous study by Montanari et al., [36], and generating the dataset required for the analysis, the following assumptions were made:

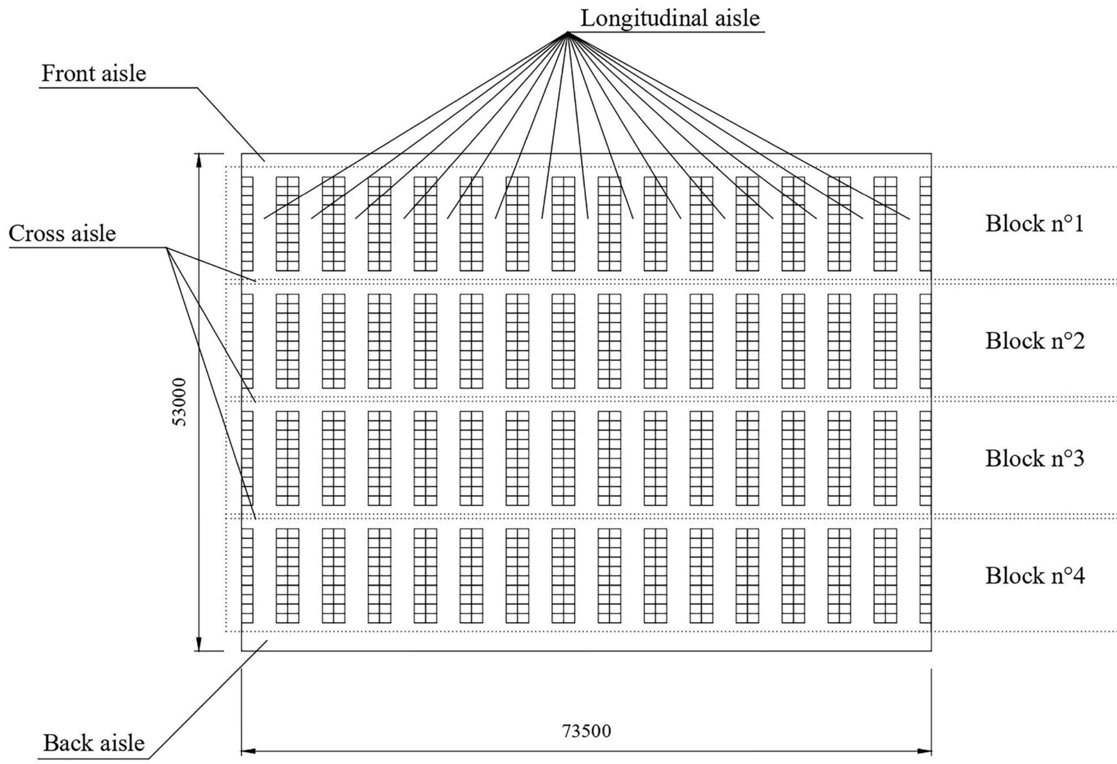

Fig. 3. Example of layout design of the warehouse with $N = 1200$, $CA=3$, $I/O=SCP$. Quotes in mm.

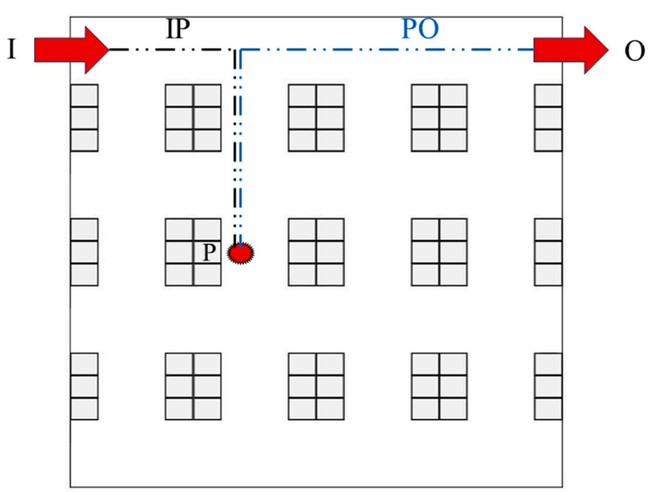

Fig. 4. Graphical representation of *R(P)*.

(1) Each location may contain one storage unit only;
(2) Each storage location contains a different product handled in the warehouse. This implicitly means that the number of products equals the storage capacity (N);
(3) The picker's route around the shelves is Manhattan metric;
(4) A manual order picking process is performed.

The input variables used embodied in the model are presented below.

2.3.1. Warehouse layout

Defining the warehouse layout means setting the elements listed below.

1) Storage capacity (N). N is set as an input parameter for defining the storage capacity of the warehouse, where each product has one ground-level storage location. By storage location, it is meant the generic spatial unit in a warehouse, in which something is stored, while a picking location is a storage location in which an item to be picked is located [9].
2) Number of cross aisles (CA). The number of CAs, in addition to the front and back aisles, defines the number of blocks into which the warehouse can be divided, as shown in Fig. 2. Whenever cross aisles are present, the warehouse is divided into a number of blocks equal to the number of cross aisles plus one [9].
3) Width of the aisles (w). This parameter allows for increasing the types of warehouses to be simulated, e.g., from an aisle allowing for the passage of a picker only, to ones that allow the passage of a traditional forklift or a trilateral one [37].
4) Size (width - α, and depth - β) of the storage location. The usage of α and β as problem variables allows for reproducing storage locations of various size, ranging from a storage area for standard EPAL pallets or ISO containers, depending on the simulation requirements. This flexibility, achieved by adjusting the α and β values, makes it possible to move from the simulation of small objects to the modelling of larger entities, expanding the adaptability of the model.
5) Shape factor (x_f). The shape factor x_f reflects the geometric characteristic of the building in which the warehouse is located. The shape factor defines the ratio between the width (a) and the depth (b) of the building, as shown in Eq. (1):

$$x_f = \frac{a}{b} \tag{1}$$

2.3.2. I/O position

By setting different I/O positions, various paths within the warehouse can be generated. Five different I/O configurations were considered, differentiated by the position of the entry and exit points (cf. [38]):

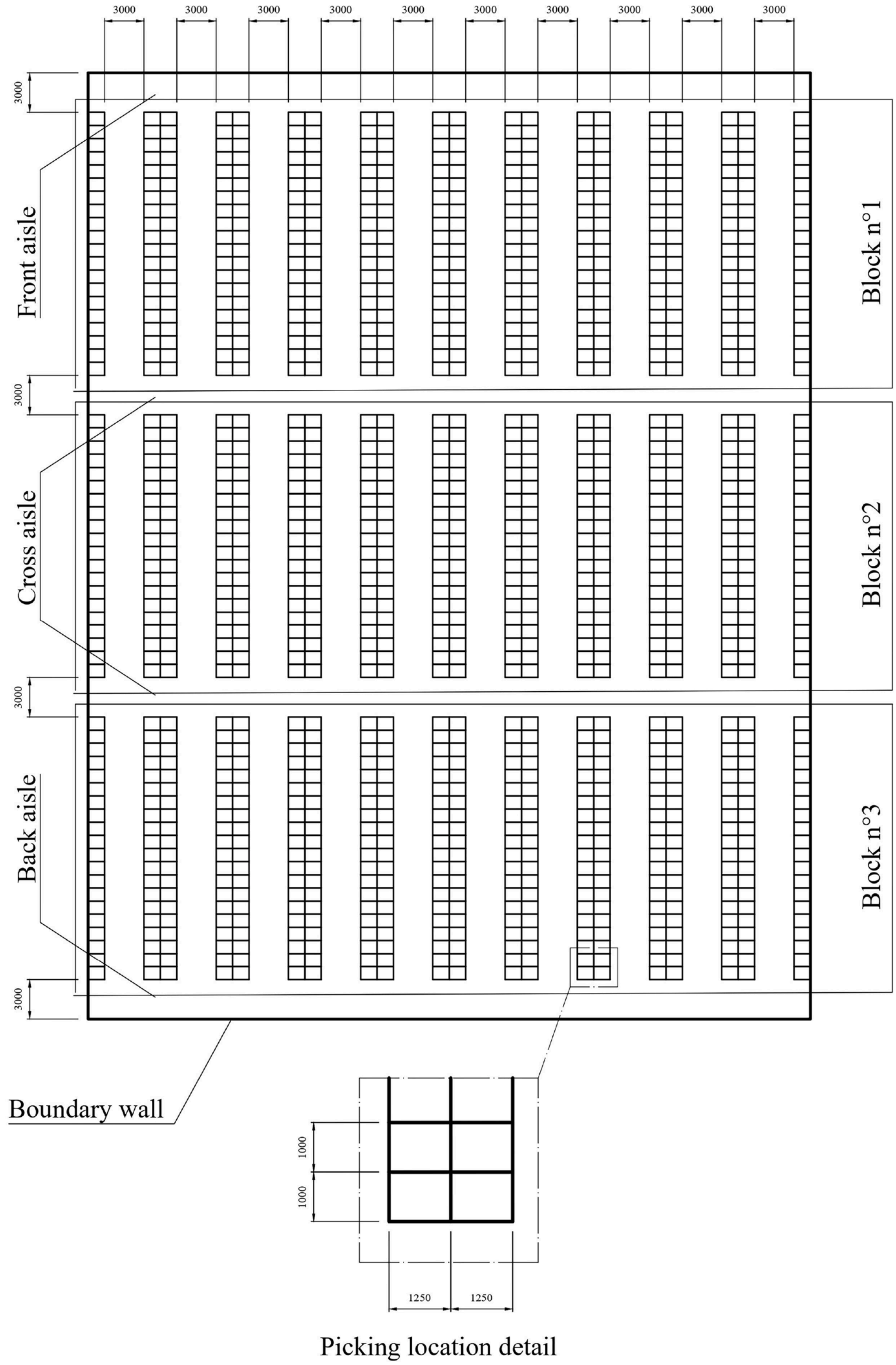

Fig. 5. 3-block warehouse layout (quotes in mm).

Table 4
Detail of the configuration chosen for analysis.

N	*α [m]*	*β [m]*	*LA*	*CA*	*a [m]*	*b [m]*	*I/O*	*w [m]*	*PL*	*λ*	*RP*
1,200	1	1.25	4	2	22	162	SCP	3	10	∼ 0	RS

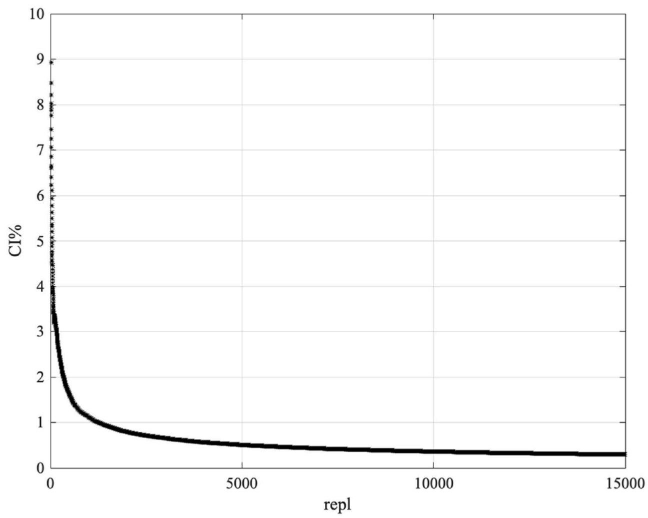

Fig. 6. *CI%* with different numbers of replications.

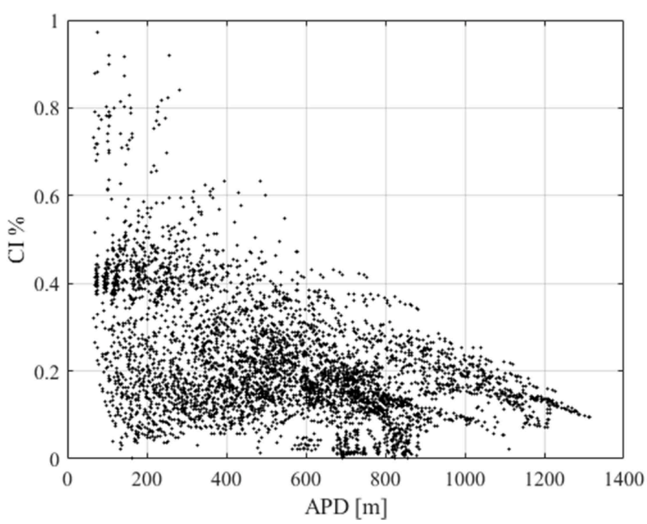

Fig. 7. *CI%* plot with a 95 % of *CL*.

Table 5
Input data for model validation.

N	*α [m]*	*β [m]*	*LA*	*CA*	*a [m]*	*b [m]*	*I/O*	*w [m]*	*PL*	*λ*
192	1	1.25	4	2	22	36	SLP	3	10	~ 0

(1) Single Central Picking (*SCP*): single central entry and exit point;
(2) Single Lateral Picking (*SLP*): single lateral entry and exit point;
(3) Opposite Lateral Picking, Same Side (*OLPSS*): the entry and exit points are positioned laterally, on the same side of the warehouse, with the entrance at the beginning and the exit at the end of the aisles;
(4) Opposite Central Picking (*OCP*): the entry and exit points are located at the beginning and the end of the aisles, respectively, in a central position;
(5) Opposite Side Picking (*OSP*): the entry and exit points are located on opposite sides of the warehouse.

2.3.3. Routing policy

The *RP* reflects the logic followed by the picker for moving in the warehouse as efficiently as possible during the picking process. Four different *RP* are evaluated and listed below [38]:

(1) Return Simple (*RS*): the picker enters and exits the aisles from the same side, collecting the items first from one side of the aisle in the entry phase, and picking from the opposite side in the exit phase. Aisles without items to be retrieved are not visited;
(2) Return Advanced (*RAD*): this policy is similar to *RS*, except that in this case the picker can also move through the cross aisles;
(3) S-Shaped Simple (*SSS*): Aisles with at least one item to be picked are entirely crossed by the picker; while traversing an aisle, the picker can collect items from the shelves on both sides. Aisles that do not contain the items on the picking list are skipped. After collecting the last item, the picker reaches the output point;
(4) S-Shaped Advanced (*SSAD*): this policy is similar to *SSS* but the picker can also move through cross aisles.

2.3.4. Demand distribution

Varying the demand distribution allows for simulating different market types, ranging, in particular, from a “uniformly distributed market”, in which each reference has the same probability of being requested by the customer, to a fully “heterogeneous market”, including both high-rotating and low-rotating items. To capture this behaviour, an appropriate probability density function (*PDF*) was defined and implemented in the simulation model (Eq. (2)):

$$PDF(i) = \frac{e^{-\lambda i}(e^{\lambda} - 1)}{1 - e^{-\lambda N}} \tag{2}$$

where:

(1) *i*: any product in the warehouse ($1 \le i \le N$);
(2) *N*: warehouse storage capacity;
(3) λ: is the parameter that allows for depicting different market scenarios. When λ tends to zero, all products are equally likely to be requested by the market, meaning that each product has the same rotation index. On the other hand, when λ increases, the demand (i.e., the market) becomes more concentrated on specific products, leading to different rotation indices for different products;
(4) *PDF(i)*: is the probability that product *i* will appear in the order list based on the market demand.

In this study, seven λ values were evaluated, which move from a uniformly distributed market $\mathrm{PDF}_{\lambda\sim0}$, up to a strongly heterogeneous market $PDF_{\lambda=0.05}$.

The Cumulative Density Function (*CDF*) for product *i* is shown in Eq. (3).

$$CDF_{\lambda}(i) = \sum_{x=1}^{i} PDF_{\lambda}(x) \tag{3}$$

Fig. 1 presents the trend of *PDF* and *CDF* for a warehouse with $N = 1200$ items. Each curve corresponds to one of the seven λ values analysed. Table 2 also shows the number of picking positions (*n*) that must be visited for each λ value, to meet at least 80 % of market demand (*CDF*).

2.3.5. Number of items in the picking list

In developing the simulator, four picking lists (*PLs*) of different lengths were taken into account as representative of typical sizes of order lists to be completed in a single picking task [26,39]; in detail, *PLs* of 10, 20, 30 and 50 items were simulated. Short *PLs* were not merged to generate larger lists [36]. To generate the picking lists, a random number was extracted from a uniform distribution in the range (0; 1] for each item to be collected; hence, the inverse function of Eq. (3) was applied as the λ parameter varied.

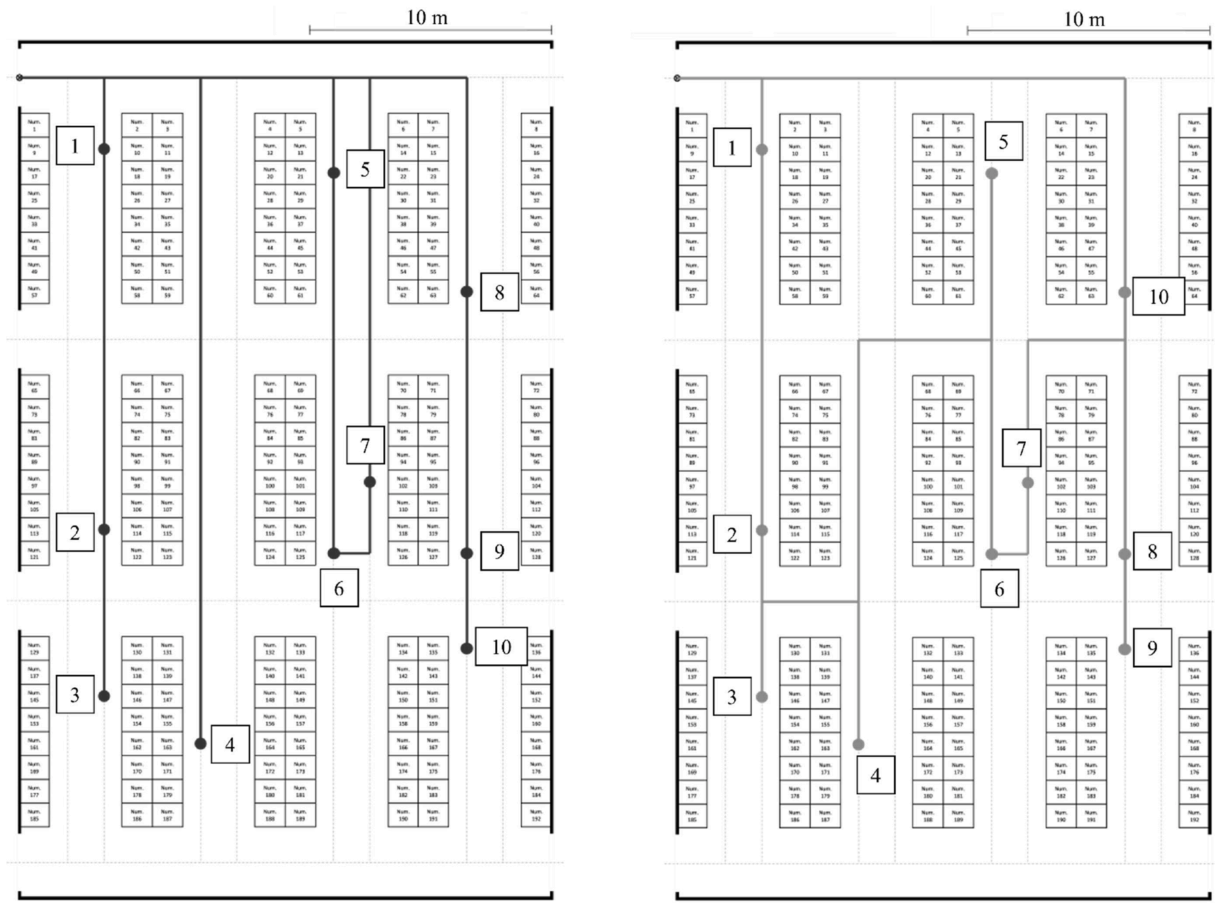

Fig. 8. Path with *RS* policy (left) and *RAD* policy (right).

With reference to the *CDF* curves in Fig. 1, it should be noted that the inverse function shows a transition from a continuous domain to a discrete codomain (Eq. (4)).

$$CDF^{-1}:\ \mathbb{R}^{+}\rightarrow\mathbb{N} \tag{4}$$

2.4. Dependent variable

In the current study, the picking distance (*PD)* was taken as the only dependent variable. In particular, for each configuration of the warehouse (*CF*), several simulation replicates were performed. Each PD_k returned by the model is a function (*f*) of the geometric tool (*g*) and the picking simulator (*h*), as shown in Eq. (5).

$$PD_k = f\left[g\left(x_f, CA, I/O, N, \alpha, \beta, w\right), h(PL, RP, \lambda)\right] \tag{5}$$

The final value of *APD* is obtained by averaging the results of the simulations performed for each *CF*, to which several random picking lists were matched.

2.5. Simulation model

A schematic representation of the simulation tool developed is presented in Fig. 2. As can be observed, the model consists of two main parts that communicate with each other during the simulation process.

The steps followed by the simulation tool (and the corresponding outcomes) are as follows:

(1) geometric representation of the warehouse;
(2) ranking of the storage location based on the calculation of their distances to *I/O* positions;
(3) definition of the product allocation according to the demand behaviour and the ranking of the storage location;
(4) simulation of the order processing, under different routing policies, with *APD* as output.

The product allocation and the ranking of the picking positions are recalculated at each simulation replicate, as they depend on both the geometry of the warehouse and the heterogeneity of market demand.

2.5.1. Geometric representation

The first part of the model, in blue, represents the geometric tool, programmed to generate a virtual model of the system, and capable of defining the storage locations of the warehouse. For reproducing the system's geometry, the tool uses the following input data:

(1) target warehouse shape factor (x_{fT});
(2) number of cross aisles (*CA*).
(3) position of the picker's entrance and exit from the warehouse (*I/O*);
(4) warehouse storage capacity (*N*);
(5) width (α) and depth (β) of the storage location;
(6) width of the aisles (*w*).

After generating the virtual model of the entire warehouse, the geometric tool calculates its size, real shape factor, and layout.

Regarding the real shape factor (x_{fR}), it is important to note that the system is obviously constrained, as integer numbers only can be used for some parameters of the warehouse. For example, it is not possible to consider fractions of allocations or corridors. Hence, a "target" x_{fT} is initially set, reflecting the expected ratio between the width and depth of

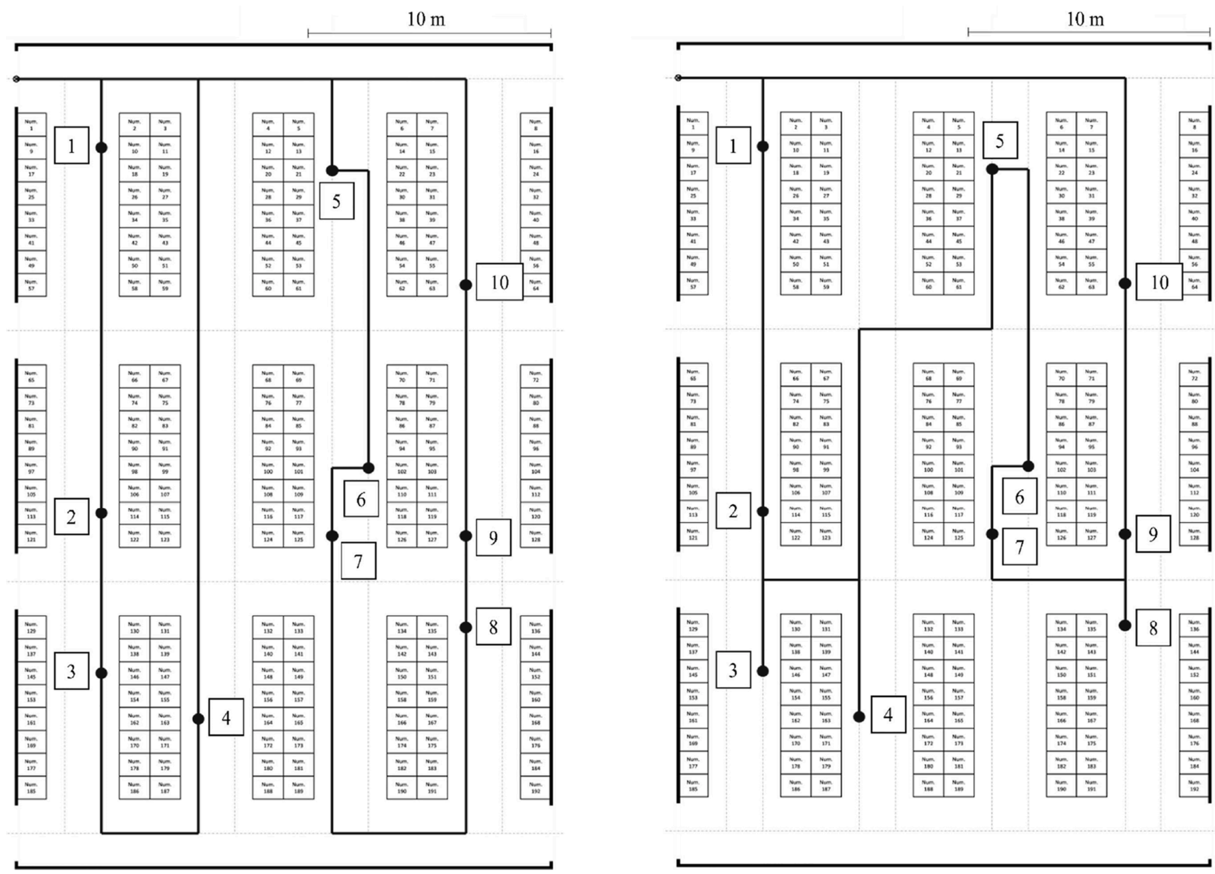

Fig. 9. Path with RS policy (left) and RAD policy (right).

Table 6
Picking distance values.

PD_{RS} [m]	PD_{RAD} [m]	PD_{SSS} [m]	PD_{SSAD} [m]
233	201	172	144

Table 7
– Variable values assumed in the analysis to generate configurations.

λ	I/O	RP	LA	PL
~ 0	SCP	RS	8	10
0.001	SLP	RAD	12	20
0.005		SSS	15	30
0.01		SSAD		50
0.02				
0.03				
0.05				

the warehouse. In generating the warehouse layout, the simulation model takes into account x_{fT}, but at the same time determines x_{fR} based on the warehouse parameters: the x_{fR} will be the one that most closely approximates x_{fT} among all possible x_f that satisfy the integer constraints (Eq. (6)). Small dissimilarities can obviously occur; hence, x_{fR} is always returned by the model for benchmarking purposes [29].

$$x_{fR} = f(N,\ CA,\ w,\ \alpha,\ \beta,\ x_{fT}) \tag{6}$$

In this study, 10 different x_{fR} are evaluated (see Table 3); these values were generated by setting $N = 1200$, $CA = 3$, $w = 3$ m, $\alpha = 1$ m and $\beta = 1.25$ m.

For the sake of clarity, Fig. 3 shows an example of a warehouse layout, in which α and β are set at 1 m and 1.2 m, respectively, w is set at 2.5 m, and x_{fT} is set at 1.5.

Based on these input values, the geometric tool returns:

- the "real" warehouse shape factor (x_{fR}) = 1.38
- the warehouse width (a) = 73.5 m
- the warehouse depth (b) = 53 m.

2.5.2. Ranking of the storage locations

Each product (i) considered in this study has its own $PDF(i)$, therefore the simulator, based on the number of *LA* and on the *I/O* positions, performs a ranking *R(P)* of each storage location (*P*), according to its distance from the entry ($\overline{IP}$) and exit ($\overline{PO}$) points (Eq. (7)). The optimal allocation will be the one that minimises the distance travelled by the picker on a picking task.

$$R(P) = \overline{IP} + \overline{PO} \tag{7}$$

A graphical representation of Eq. (7) is shown in Fig. 4 for an OSP I/O configuration.

It should be noted that the *R(P)* value does not reflect the Euclidean distance but must account for the presence of geometric constraints, i.e., the presence of shelves and the position of the cross aisles.

2.5.3. Definition of the product allocation

The second part of the simulation tool, in black in Fig. 2, is the picking simulator and receives as input the following data:

(1) picking list (*PL*);
(2) routing policy (*RP*);
(3) coefficient related to demand distribution (λ).

This part of the simulator communicates with the first section,

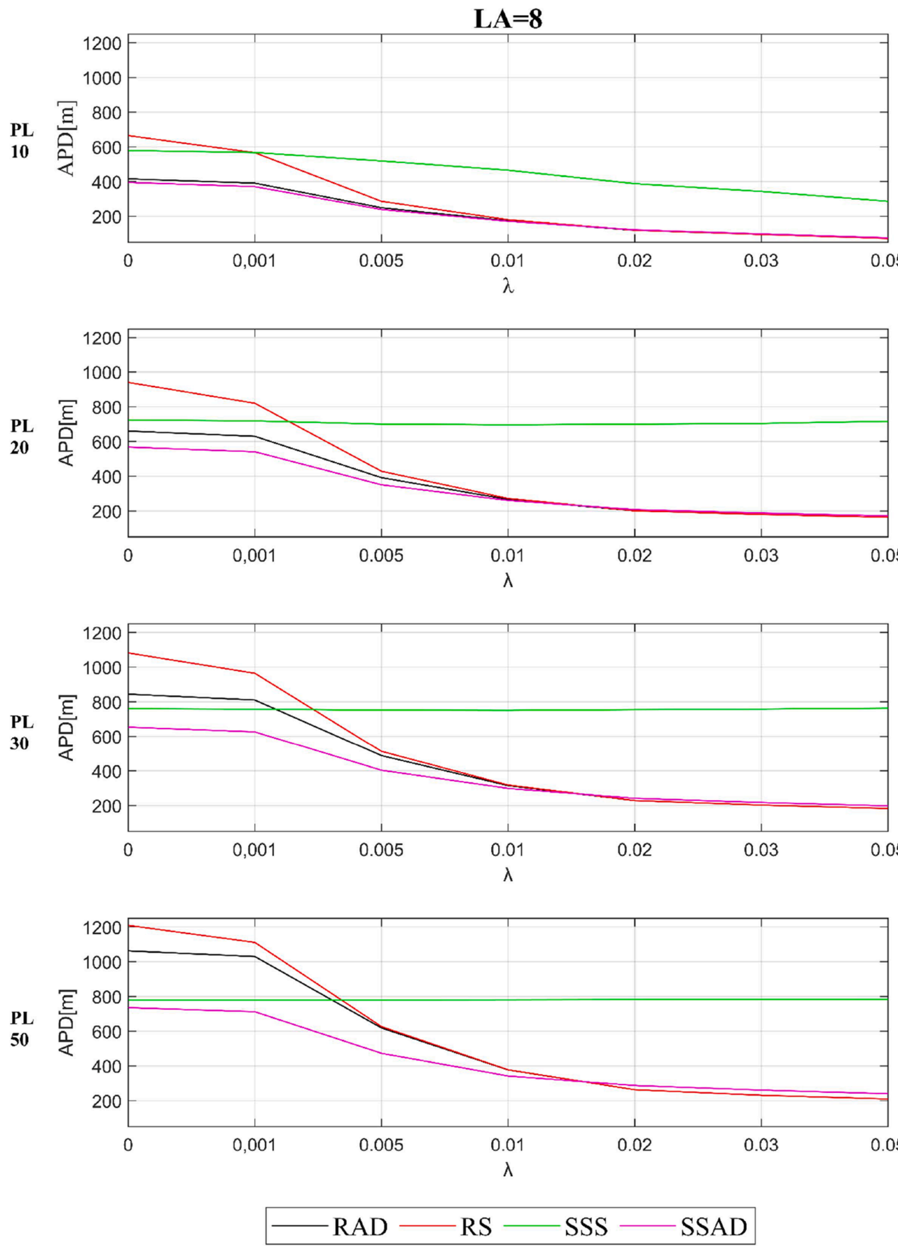

Fig. 10. Plotting Single Central Picking configuration and LA=8.

described above, as any variation in the market demand obviously influences the allocation of items within the warehouse. This allows for generating an output, in terms of *APD*, that takes into consideration both hardware and software constraints of the warehouse under examination.

In particular, the simulator matches the items in decreasing order of the $PDF(i)$ with the storage location (from the most favourable location to the worst one), based on the previously calculated value of *R(P)* Eqs. (8)-(10).

$$\widetilde{R}(P) = R(P),\ 1 \le P \le N|R(P+1) > R(P) \tag{8}$$

$$\widetilde{PDF}_{\lambda}(i) = PDF_{\lambda}(i),\ 1 \le i \le N|PDF_{\lambda}(i+1) < PDF_{\lambda}(i) \tag{9}$$

$$\widetilde{R}(P) \rightarrow \widetilde{PDF}_{\lambda}(i)|P = i \tag{10}$$

Depending on the hardware configuration set, the application of the above formulae enables the complete mapping of all storage locations. Accordingly, the item with the highest probability of being requested by the market will be placed in the most favourable position, while the item with the lowest probability of being requested will be placed in the most unfavourable position.

2.5.4. *Simulation of the order processing*

In this analysis, a traditional warehouse layout with double-sided racking was modelled, with N = 1,200 storage locations. These are characterized by $\alpha = 1$ m and $\beta = 1.25$ m, while *CA* and *LA* have $w = 3$ m. In addition to the front and rear aisles, there are also two *CA* in the warehouse, resulting in a 3-block layout (Fig. 5). Considering these assumptions, a full factorial plan was used to simulate all the possible combinations of the independent variables, resulting in a total of 5,600 simulated configurations, as reported in Eq. (11).

$$CF = n^{\circ}RP \cdot n^{\circ}x_f \cdot n^{\circ}I/O \cdot n^{\circ}PL \cdot n^{\circ}PDF = 5,600 \tag{11}$$

2.6. *Statistical analysis*

To provide statistical significance to the outcomes of Eq. (5), a

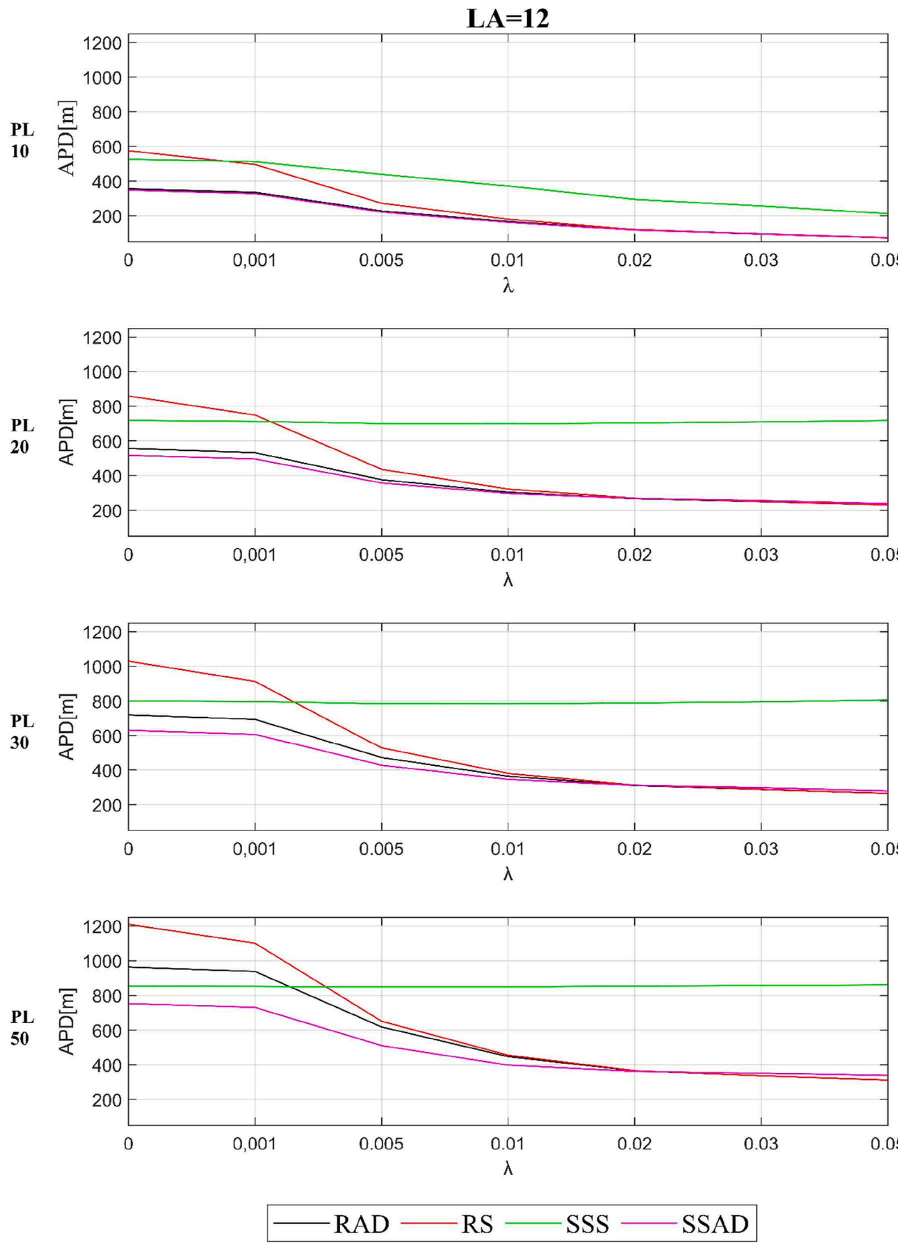

Fig. 11. Plotting Single Central Picking configuration and LA=12.

suitable number of replicates (*repl*) to be adopted was determined (Eq. (12)):

$$APD = \frac{\sum_{k=1}^{repl} PD_k}{repl} \tag{12}$$

In this analysis, assuming a normal distribution of the values, the confidence interval (*CI*) will be defined by the *t-Student* distribution. Eq. (13) is used to define the *CI*, adopting a *t*-value according to a 95 % confidence level (*CL*).

$$CI = \frac{\sigma_{APD}}{\sqrt{repl}} \times t_{[(1-CL)/2,\ repl-1]} \tag{13}$$

For testing purpose, this procedure was applied to a specific configuration of the database under investigation, whose characteristics are shown in Table 4.

In the present case, a configuration with the lowest values of both *PL* and λ was chosen to validate the analysis. Hence, Eq. (13) was applied to the configuration shown in Table 4 with different levels of replicates (from 20 to 15,000). Fig. 6 shows the different values of the resulting confidence intervals (*CI%*) as the number of replicates increases.

The simulation model was developed by implementing VBA code within the Excel™ software, run on a machine with an Intel® Core™ i9–10885H CPU @2.40 GHz processor and a RAM of 64 GB. The computational time for each configuration was 6.63 $\times 10^{-9}$ s.

As can be seen from Fig. 6, with a number or replicates equal to 10,000 and a *CL* of 95 %, the *CI%* turns out to be 0.36 %, corresponding to an *APD* value equal to 856,31 $m \pm 3.071$ m; no tangible benefits emerge if further increasing the number of replicates. On the basis of this result, a number of 10,000 replicates was judged adequate for the scope of the paper.

The *CI%* values of the whole dataset plotted against their *APD* are shown for completeness in Fig. 7.

As can be seen from the graph in Fig. 7, by performing 10,000 simulations for each configuration, the confidence interval for a 95 % confidence level of the values is always less than 1 % of the *APD*.

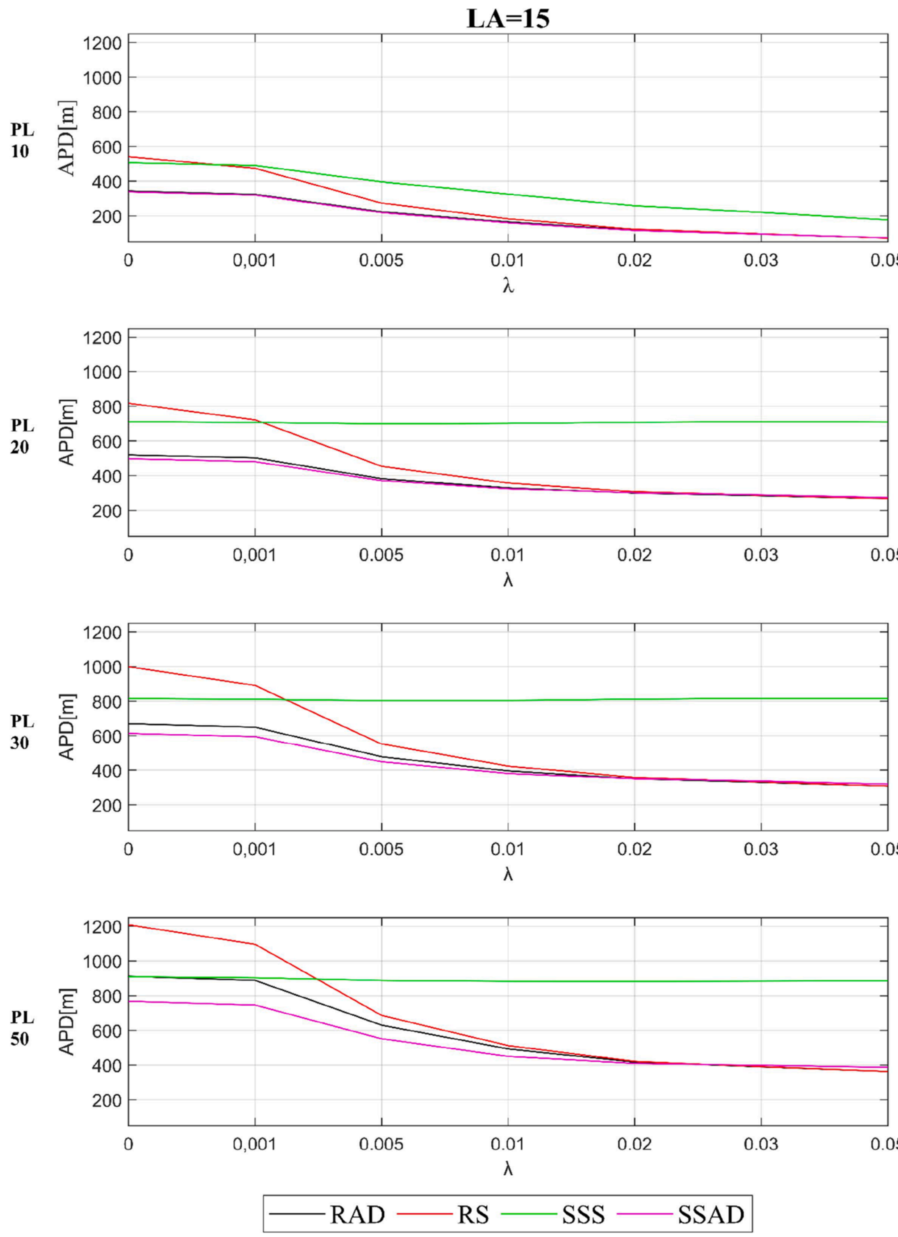

Fig. 12. Plotting Single Central Picking configuration and LA=15.

3. Results

3.1. Model validation

This section provides a validation of the model outlined above. This is shown below for a simplified warehouse configuration, whose input data are listed in Table 5.

A random *PL* of 10 items was generated and the four different routing policies were applied to pick up these items from the warehouse. Each *RP* generates a different path, these picking paths (*PD*) are shown in Fig. 8 for *RS* and *RAD* policies and Fig. 9 for *SSS* and *SSAD* policies respectively.

Table 6 shows the *PD* values with the 4 different policies described above.

3.2. Investigation boundaries

In analysing the results, particular attention was paid to a part of the dataset, to investigate:

- two I/O configurations: *SCP* and *SLP*;
- three values of *LA*: 8 (x_{fR}= 0.51); 12 (x_{fR}= 1.06); 15 (x_{fR}= 1.59).

The rationale for limiting the analysis to these aspects is that of using hardware parameters (*LA* and *I/O* values) typically observed in real scenarios, which increases the likelihood of investigating warehouse configurations suitable for practical implementation.

The remaining input parameters were instead fully evaluated, as reported in Table 7. By including all λ values, several possible demand scenarios were analysed, from a uniformly distributed to a strongly sectorised demand.

By combining the set of parameters in Table 7, the total number of configurations analysed turns out to be 7·2·4·3·4 = 672.

3.3. SCP and SLP

Figs. 10-12 and Figs. 13-15 refer to the *SCP* and *SLP* I/O configurations, respectively. In each figure, corresponding to a single *LA* value, there are four plots, corresponding to the 4 *PLs*. In each graph, *APD* is

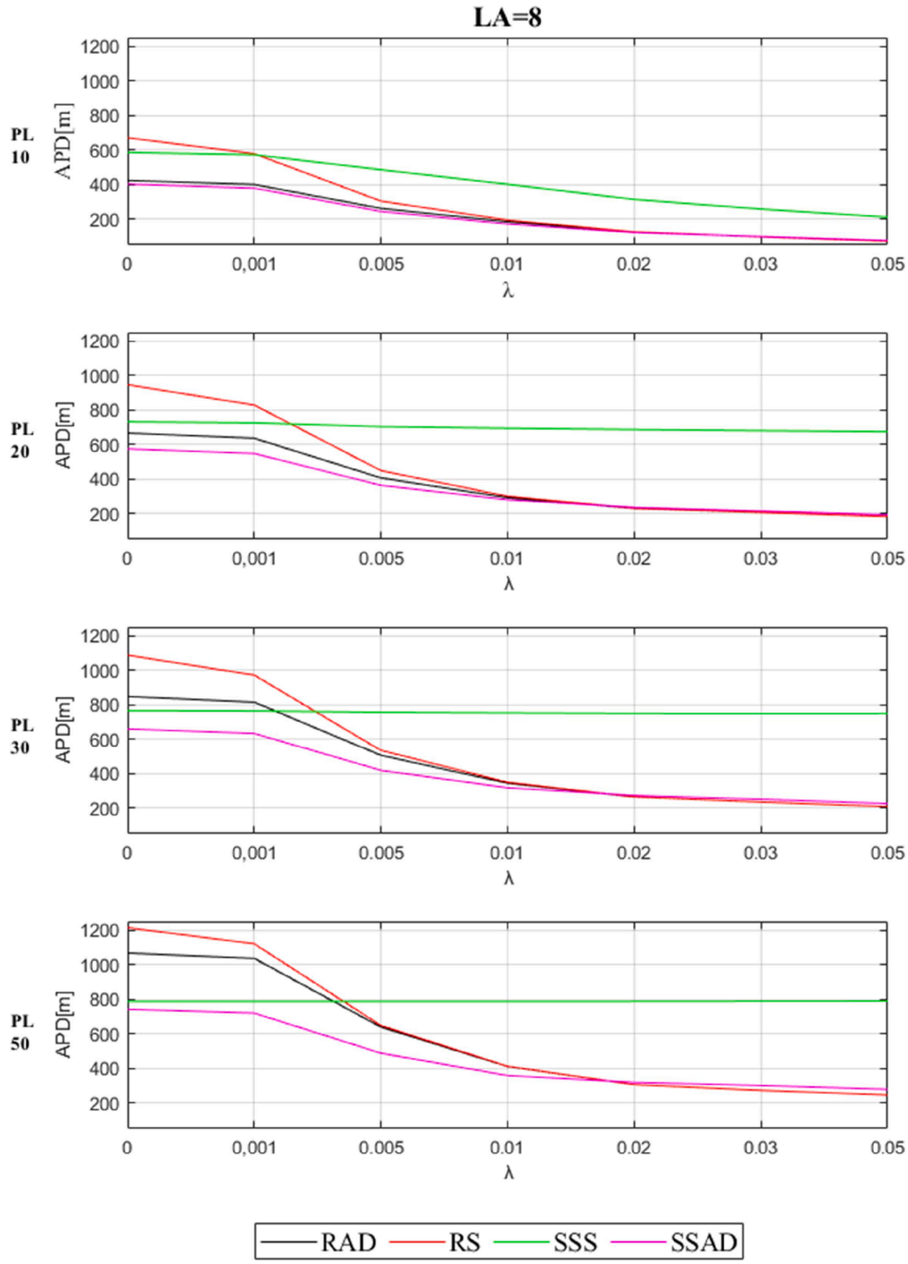

Fig. 13. Plotting Single Lateral Picking configuration and LA=8.

represented as a function of the λ value. In addition, four curves in each graph depict the four *RP* evaluated in this study (i.e., *RS, RAD, SSS, SSAD*).

From an initial analysis, it emerges that when applying the *SSS* policy, *APD* is weakly influenced by variations in the λ parameter, except in the case of very small *PLs* (10 items); in that case, a slight dependence on λ can be observed, albeit lower than the remaining policies. A justification for this behaviour is to be found in the functioning of the *SSS* logic itself, in which the picker, once entering an aisle to pick an item, is forced to travel it entirely and exit on the opposite side. Consequently, having a *PL* with a wide number of items increases the probability of covering the entire warehouse during picking travel. Even if λ increases, i.e., when considering high-rotating and low-rotating articles, no significant improvements in the performance can be observed. Indeed, it is true that the items with the highest demand will be placed close to the *I/O* point, but this also means that they will be placed as close as possible to the front aisles, and consequently, they will fall in different *LA*s; hence, again when using the *SSS* policy, no remarkable reduction in the *APD* can be observed. These outcomes reinforce the consideration that, in general, the *S-shaped* routing policy is not particularly effective [7]; this is also true in contexts in which the demand patterns can vary.

An interesting result also arises from the previous plots: apart from the *SSS* policy analysed above, it can be observed that for $\lambda \geq 0.01$, the *RS, RAD* and *SSAD* policies show similar outcomes in terms of *APD*. This outcome complements the available knowledge by highlighting that in some contexts (although somehow difficult to observe in real scenarios as market demand is highly heterogeneous), these policies could be considered interchangeable, and the usage of any of them would not affect the performance of the picking process. Ultimately, this allows good flexibility in the choice of the *RP*, which can be set according to the specific needs and constraints of the application context.

The *APD* values of the *RS, RAD* and *SSAD* routing policies with $\lambda \geq 0.01$ are summarised in Tables 8 and 9 for the two selected *I/O* configurations. The Δ value, intended as the percentage deviation between the maximum and minimum *APD* values (Eqs. (14)-16), has also been added to these two tables for a more effective comparison of the

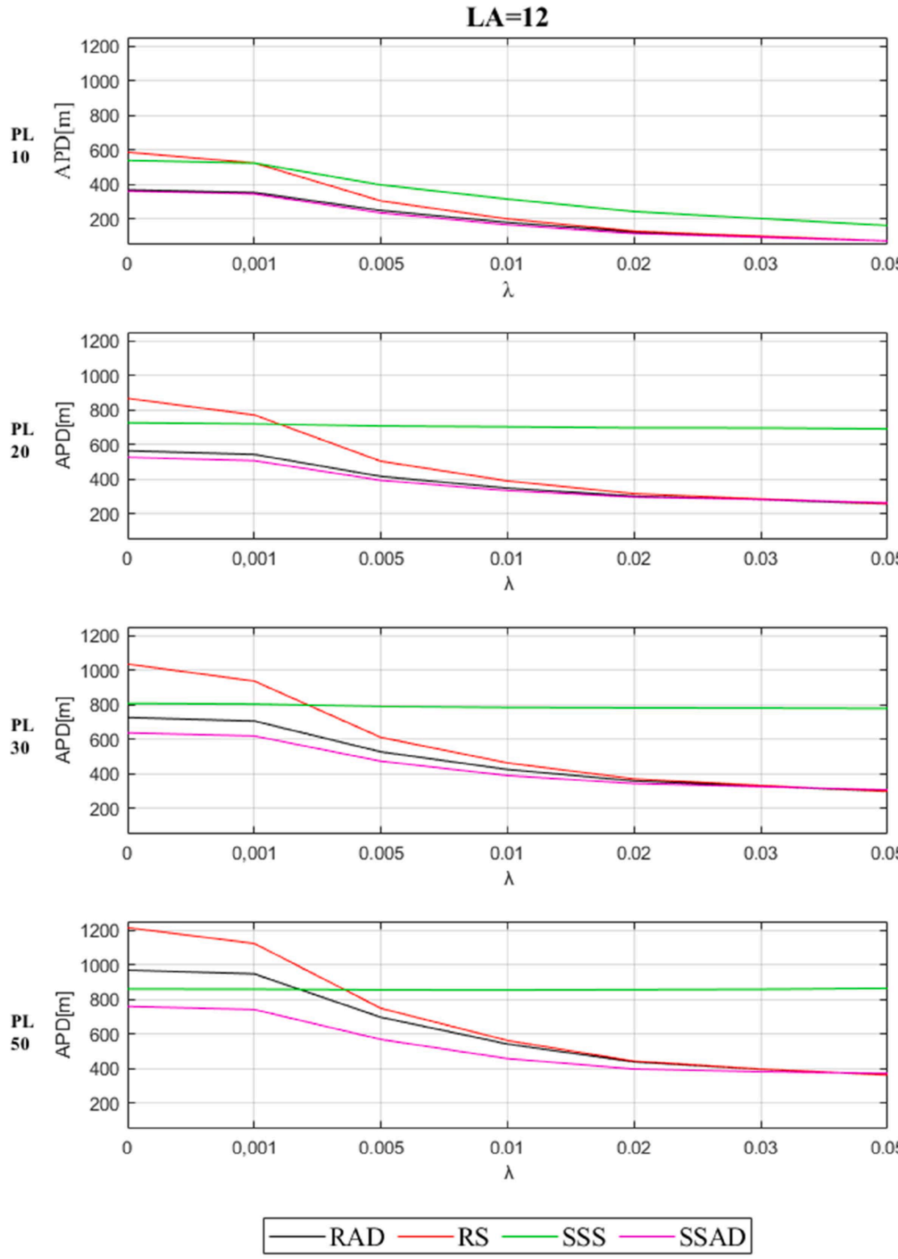

Fig. 14. Plotting Single Lateral Picking configuration and *LA*=12.

routing policies:

$$M = \max\left[\left(APD|_{RP=RS}\right), \left(APD|_{RP=RAD}\right), \left(APD|_{RP=SSAD}\right)\right] \tag{14}$$

$$m = \min\left[\left(APD|_{RP=RS}\right), \left(APD|_{RP=RAD}\right), \left(APD|_{RP=SSAD}\right)\right] \tag{15}$$

$$\Delta = \frac{M-m}{m} \cdot 100 \tag{16}$$

The Δ parameter allows for evaluating the range of *APD* values as the *RP* varies; $\Delta \leq 10$ % was considered a small deviation.

Concerning the *SCP* configuration, it can be seen that only 9 scenarios out of 144 considered were characterized by $\Delta > 10$ %, while in the case of *SLP,* 18 scenarios out of 144 had $\Delta > 10$ %.

Among the policies analysed, *RS* differs from *RAD* and *SSAD* in the fact that it does not make use of either the *CA* or the corridor at the bottom. This is an important aspect to consider in specific operational contexts, where the removal of such aisles could favour an increase in warehouse capacity or a decrease in the area occupied. These variations in the layout could be implemented without compromising the necessary storage capacity nor the efficiency of the picking process in terms of *APD*. Moreover, *RS* is a very simple policy and can be considered the easiest logic to be understood and implemented by a picker. Consequently, when adopting this policy, it will also be easier to obtain a simple route for the picker to take during the order processing; the possibility of incorrect routing is therefore reduced, with a benefit on the overall process efficiency and effectiveness.

On the other hand, looking again at Figs. 10-12 and Figs. 13-15, it is easy to see that for scenarios with $\lambda < 0.01$ (which are more likely to be observed in real environments), the best routing policy overall is the *SSAD*, as demonstrated by the *APD* value shown in Tables 10,11.

One of the most interesting findings of the study, in the authors' opinion, concerns the impact of the heterogeneity in the market demand on the choice of the optimal routing policy. Other important outcomes concern the difference in that impact as a function of the *RP* considered; indeed, depending on the heterogeneity of demand, different *RPs* can be identified as optimal, which ultimately, can influence the hardware

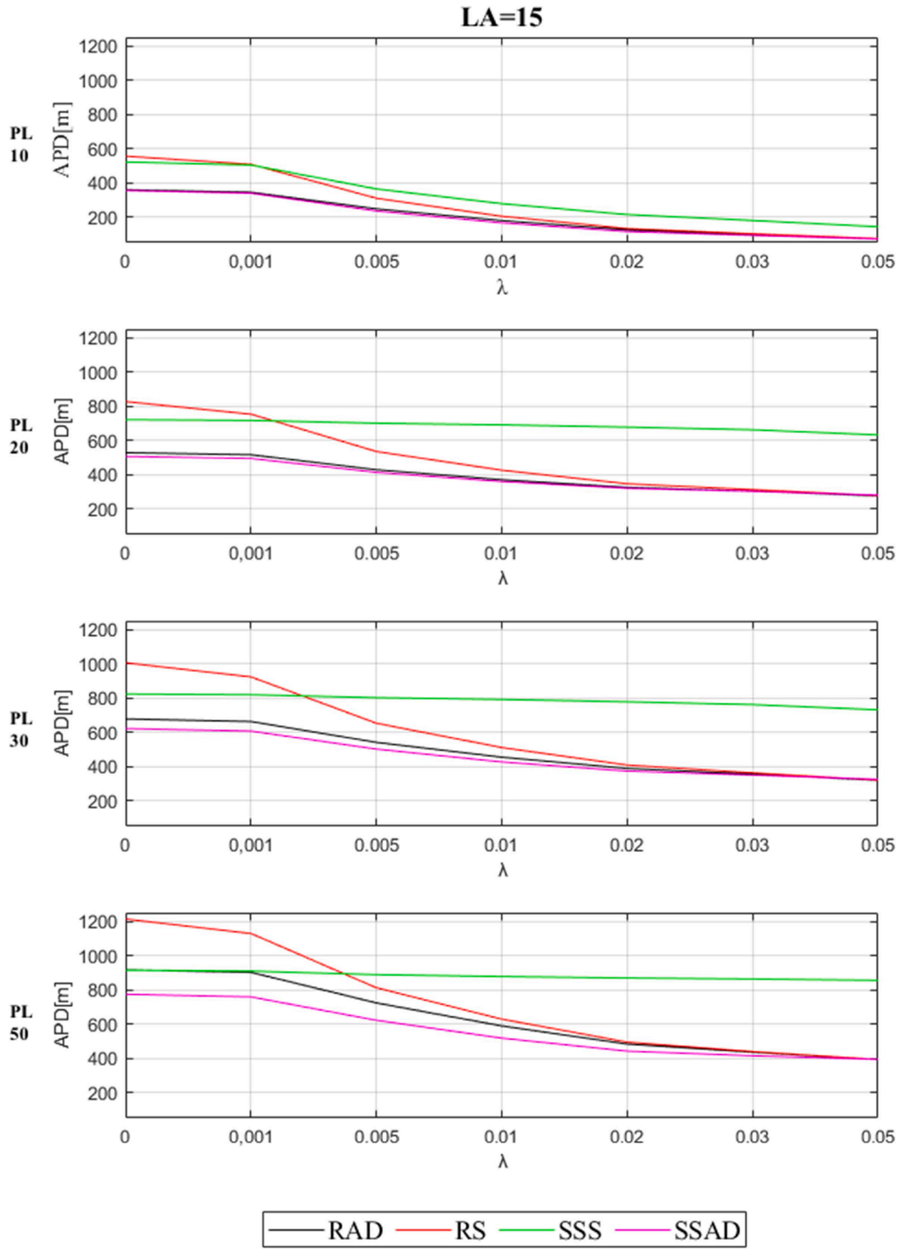

Fig. 15. Plotting Single Lateral Picking configuration and LA=15.

characteristics of the warehouse. This line of reasoning is summarised Fig. 16. In particular, with $\lambda \geq 0.01$ (right side of the graph), the *RS* policy turns out to be optimal; as that policy does not make use of *CAs*, their presence will not affect the performance of the picking process and, consequently, *CAs* could be removed from the warehouse. On the contrary, with $\lambda < 0.01$ (left side of the graph), the implementation of policies which instead make use of cross aisles (*SSAD*) increases the performance of the warehouse, being the best policy in all scenarios considered for these demands from the market.

4. Discussion and conclusions

In the present study, a simulation tool was created to reproduce a manual picking process in a traditional double-side rack warehouse. The ultimate objective of the study was to investigate different warehouse configurations, by varying both software (*RP, PL,* λ) and hardware (x_f, *CA, w, N,* α, β, *I/O*) parameters of the system, exploiting the simulation tool, and to analyse their impact on the resulting *APD*.

In line with the approach by Manuj et al. [30], the starting point for developing the model was the problem formulation, followed by the definition of the independent and dependent variables. Then, a simulation campaign was carried out using the model, with results averaged on 10,000 replicates for each configuration considered. During each simulation, the picking lists were varied.

By analysing a set of selected scenarios out of the whole set of data obtained through simulation, two key results relating to the *RPs* were observed. In the case of low λ values, reflecting a homogeneous demand of items (and implicitly, random storage), the best policy turned out to be *SSAD*. This result corroborates the findings by Bottani et al. [29], who reported that the *SSAD* policy is effective under various warehouse configurations. However, with high λ values, i.e., strongly heterogeneous demand, *SSAD, RAD* and *RS* all exhibit comparable performance. This outcome has various implications. First, it highlights that the heterogeneity in demand, as modelled in this study, affects the picker routing, and thus, this aspect is to be taken into account when trying to optimize the order picking process. As a second point, the fact that more

Table 8
APD [m] values in *SCP* configuration for $\lambda \geq 0.01$.

	LA=8						*LA*=12						*LA*=15					
PL= 10			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**
	λ	0.01	180	174	171	**5.3 %**	λ	0.01	181	167	163	**11.2 %**	λ	0.01	184	165	160	**14.9 %**
		0.02	119	119	122	**2.1 %**		0.02	121	118	117	**2.8 %**		0.02	123	118	115	**6.6 %**
		0.03	95	95	99	**3.6 %**		0.03	96	95	97	**1.5 %**		0.03	96	95	94	**2.2 %**
		0.05	72	72	76	**5.2 %**		0.05	72	72	75	**4.5 %**		0.05	72	72	74	**3.0 %**
PL=20			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**
	λ	0.01	272	266	260	**4.3 %**	λ	0.01	323	304	298	**8.4 %**	λ	0.01	358	330	325	**10.0 %**
		0.02	201	201	208	**3.6 %**		0.02	270	267	269	**1.4 %**		0.02	307	300	301	**2.5 %**
		0.03	180	180	188	**4.5 %**		0.03	251	250	255	**2.2 %**		0.03	288	285	290	**1.5 %**
		0.05	164	164	172	**5.0 %**		0.05	232	231	239	**3.2 %**		0.05	268	268	275	**2.4 %**
PL=30			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**
	λ	0.01	318	315	299	**6.4 %**	λ	0.01	380	364	345	**10.1 %**	λ	0.01	424	395	381	**11.3 %**
		0.02	229	229	241	**5.6 %**		0.02	313	310	312	**1.0 %**		0.02	358	351	351	**2.1 %**
		0.03	202	202	217	**7.3 %**		0.03	288	288	297	**3.3 %**		0.03	333	330	337	**2.1 %**
		0.05	183	183	198	**8.4 %**		0.05	266	266	279	**5.1 %**		0.05	308	307	320	**4.0 %**
PL=50			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**
	λ	0.01	378	377	342	**10.5 %**	λ	0.01	455	447	398	**14.3 %**	λ	0.01	512	493	450	**13.7 %**
		0.02	263	263	287	**9.2 %**		0.02	366	365	363	**1.0 %**		0.02	422	418	410	**2.8 %**
		0.03	231	231	261	**12.9 %**		0.03	337	337	351	**4.3 %**		0.03	391	390	399	**2.3 %**
		0.05	210	210	240	**14.5 %**		0.05	313	313	338	**8.2 %**		0.05	364	364	386	**6.2 %**

Table 9
APD [m] values in *SLP* configuration for $\lambda \geq 0.01$.

	LA=8						*LA*=12						*LA*=15					
PL= 10			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**
	λ	0.01	193	184	171	**12.9 %**	λ	0.01	199	179	165	**20.5 %**	λ	0.01	204	178	165	**23.5 %**
		0.02	125	124	121	**3.4 %**		0.02	129	124	115	**12.3 %**		0.02	132	124	114	**15.3 %**
		0.03	97	97	99	**1.7 %**		0.03	99	98	93	**7.0 %**		0.03	102	99	92	**10.7 %**
		0.05	71	71	76	**6.6 %**		0.05	72	72	73	**1.0 %**		0.05	73	73	71	**3.6 %**
PL=20			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**
	λ	0.01	299	291	279	**7.2 %**	λ	0.01	389	347	332	**16.9 %**	λ	0.01	425	370	359	**18.4 %**
		0.02	230	229	235	**2.6 %**		0.02	314	301	296	**6.3 %**		0.02	346	325	320	**8.2 %**
		0.03	206	206	215	**4.4 %**		0.03	285	280	280	**2.0 %**		0.03	312	303	302	**3.4 %**
		0.05	182	182	193	**5.5 %**		0.05	257	256	262	**2.4 %**		0.05	278	276	280	**1.6 %**
PL=30			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**
	λ	0.01	348	343	316	**10.0 %**	λ	0.01	462	424	389	**18.8 %**	λ	0.01	510	454	426	**19.7 %**
		0.02	264	264	273	**3.4 %**		0.02	369	358	343	**7.7 %**		0.02	408	388	373	**9.4 %**
		0.03	234	234	250	**6.6 %**		0.03	332	328	325	**2.0 %**		0.03	362	355	349	**3.8 %**
		0.05	208	208	225	**8.6 %**		0.05	298	297	307	**3.3 %**		0.05	320	319	324	**1.9 %**
PL=50			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**			*RS*	*RAD*	*SSAD*	**Δ**
	λ	0.01	411	410	358	**15.0 %**	λ	0.01	561	541	457	**22.9 %**	λ	0.01	628	588	517	**21.5 %**
		0.02	305	305	318	**4.4 %**		0.02	442	437	396	**11.5 %**		0.02	494	483	441	**11.8 %**
		0.03	272	272	300	**10.3 %**		0.03	396	395	380	**4.3 %**		0.03	439	435	414	**5.9 %**
		0.05	245	245	278	**13.3 %**		0.05	362	362	371	**2.6 %**		0.05	392	392	393	**0.3 %**

policies return the same travel distance also indicates that, with a correct item allocation, the usage of a specific routing policy becomes indifferent. Ultimately, this introduces the consideration that if products have markedly different behaviours, a correct item allocation is to be prioritized over the choice of the routing policy. This is an important practical aspect, as changing the allocation of items in the warehouse is a costly activity, which takes days of work and cannot be made frequently; therefore, carefully evaluating the allocation of the items and keeping it unchanged is critical to the efficiency of the system. Additional analyses are nonetheless recommended for the future to investigate this point in greater detail. As far as the three policies with similar behaviour, again from a practical perspective, it could be reasonable to privilege the implementation of the *RS* logic, which not only generates very easy routes for the picker but also does not require the *CA*. Although not evaluated in this study, it is known that an easier route often corresponds to a lower probability of human errors, which further favours the implementation of the *RS* policy. The *SSS* policy is instead an exception and shows an anomalous behaviour; by the way, researchers have already indicated that this policy is unlikely to be effective (e.g., [7]), as the picker is always forced to traverse the whole corridor to pick the items of the picking list, resulting in longer routes [29].

Building on this work, it is recommended to take future research activities focusing on the study of other *I/O* configurations, the comparative evaluation of the impact of scattered storage vs. dedicated storage, or, as mentioned above, the quantitative evaluation of the impact of human errors on the efficiency of the order picking process. These analyses could be effectively supported by the simulation tool presented in this study.

Table 10
APD [m] values in *SCP* configuration for λ<0.01.

	LA=8						*LA*=12						*LA*=15					
PL= 10			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*
	λ	~ 0	665	416	579	395	λ	~ 0	576	356	526	348	Λ	~ 0	541	343	506	339
		0.001	566	391	567	371		0.001	496	336	512	329		0.001	474	324	490	320
		0.005	286	249	519	239		0.005	272	227	439	222		0.005	273	223	396	219
PL=20			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*
	λ	~ 0	940	660	724	567	λ	~ 0	859	556	718	517	λ	~ 0	817	520	711	497
		0.001	820	630	718	541		0.001	749	532	712	495		0.001	722	502	707	480
		0.005	429	392	700	350		0.005	436	376	699	358		0.005	454	383	698	372
PL=30			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*
	λ	~ 0	1082	844	760	654	λ	~ 0	1030	720	802	631	λ	~ 0	999	671	816	614
		0.001	965	810	758	626		0.001	913	694	797	607		0.001	891	650	812	595
		0.005	513	488	752	403		0.005	528	472	784	427		0.005	552	478	804	448
PL=50			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*
	λ	~ 0	1209	1062	780	735	λ	~ 0	1210	964	854	753	λ	~ 0	1208	912	909	768
		0.001	1111	1030	780	712		0.001	1101	938	852	730		0.001	1096	888	904	746
		0.005	626	619	780	473		0.005	651	618	849	511		0.005	686	630	888	552

Table 11
APD[m] values in *SLP* configuration for λ<0.01.

	LA=8						*LA*=12						*LA*=15					
PL= 10			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*
	λ	~ 0	670	423	586	402	λ	~ 0	585	368	538	361	λ	~ 0	555	358	521	354
		0.001	578	400	572	378		0.001	523	353	522	344		0.001	507	344	504	338
		0.005	304	261	485	244		0.005	304	248	396	234		0.005	309	247	363	235
PL=20			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*
	λ	~ 0	945	665	730	574	λ	~ 0	866	563	726	525	λ	~ 0	827	528	721	506
		0.001	827	635	724	547		0.001	771	542	719	506		0.001	753	516	716	494
		0.005	448	406	703	363		0.005	503	415	706	392		0.005	535	428	700	413
PL=30			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*
	λ	~ 0	1086	847	766	659	λ	~ 0	1035	725	808	636	λ	~ 0	1005	677	823	621
		0.001	971	814	764	633		0.001	936	705	803	618		0.001	923	662	819	606
		0.005	534	505	756	418		0.005	610	526	789	472		0.005	652	540	801	501
PL=50			*RP*						*RP*						*RP*			
			RS	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*			*RS*	*RAD*	*SSS*	*SSAD*
	λ	~ 0	1213	1066	786	741	λ	~ 0	1215	968	859	758	λ	~ 0	1213	917	915	774
		0.001	1120	1035	786	719		0.001	1123	947	858	741		0.001	1129	902	909	758
		0.005	647	638	786	487		0.005	748	696	854	567		0.005	812	723	888	622

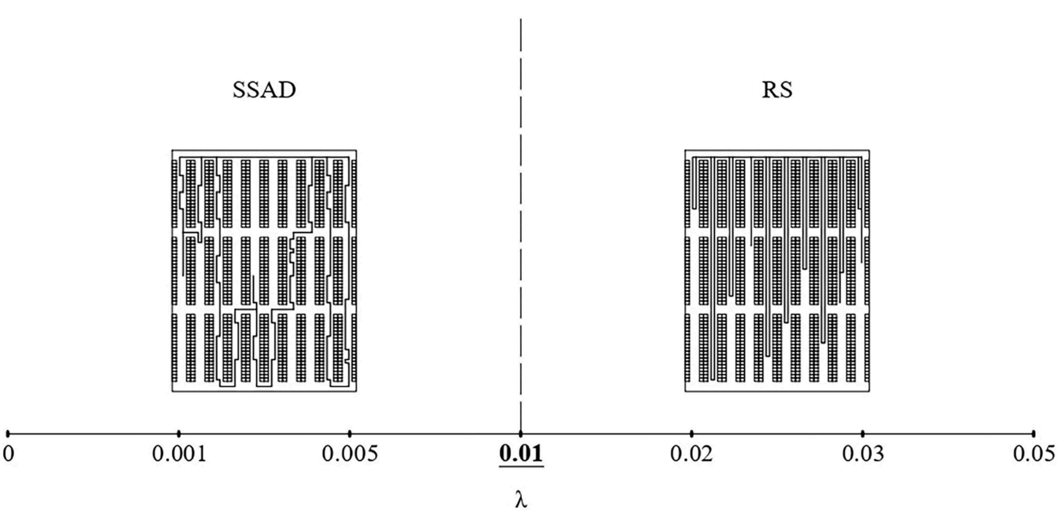

Fig. 16. Optimal routing policy based on λ value.

CRediT authorship contribution statement

Michele Bocelli: Writing – review & editing, Writing – original draft, Validation, Formal analysis. **Eleonora Bottani:** Writing – review & editing, Writing – original draft, Methodology. **Andrea Volpi:** Software, Investigation. **Federico Solari:** Software, Conceptualization. **Natalya Lysova:** Visualization, Data curation. **Roberto Montanari:** Supervision, Project administration, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] M. Amorim-Lopes, L. Guimarães, J. Alves, B. Almada-Lobo, Improving picking performance at a large retailer warehouse by combining probabilistic simulation, optimization, and discrete-event simulation, Int. Trans. Oper. Res. 28 (2) (2021) 687–715, https://doi.org/10.1111/itor.12852.

[2] J. Zhang, X. Zhang, Y. Zhang, A study on online scheduling problem of integrated order picking and delivery with multizone vehicle routing method for online-to-offline supermarket, Math. Probl. Eng. (2021), https://doi.org/10.1155/2021/6673079. *2021*.

[3] B. Aerts, T. Cornelissens, K. Sörensen, The joint order batching and picker routing problem: modelled and solved as a clustered vehicle routing problem, Comput. Oper. Res. 129 (2021) 129, https://doi.org/10.1016/j.cor.2020.105168.

[4] J. Tompkins, J. White, Y. Bozer, E. Frazelle, J. Tanchoco, J. Trevin, Facilities Planning, Wiley, New York, 1996.

[5] R. Hall, Distance approximations for routing manual pickers in a warehouse, IIE Trans. 25 (4) (1993) 76–87, https://doi.org/10.1080/07408179308964306.

[6] J. Jarvis, E. McDowell, Optimal product layout in an order picking ware- house, IIE Trans. 23 (1) (1991) 93–102, https://doi.org/10.1080/07408179108963844.

[7] C. Petersen, The impact of routing and storage policies on warehouse efficiency, Int. J. Oper. Prod. Manag. 19 (10) (1999) 1053–1064, https://doi.org/10.1108/01443579910287073.

[8] C. Petersen, G. Aase, A comparison of picking, storage, and routing policies in manual order picking, Int. J. Prod. Econ. 92 (1) (2004) 11–19, https://doi.org/10.1016/j.ijpe.2003.09.006.

[9] K. Roodbergen, R. de Koster, Routing methods for warehouses with multiple cross aisles, Int. J. Prod. Res. 39 (9) (2001) 1865–1883, https://doi.org/10.1080/00207540110028128.

[10] E. Bottani, A. Volpi, R. Montanari, Design and optimization of order picking systems: an integrated procedure and two case studies, Comput. Ind. Eng. 137 (2019), https://doi.org/10.1016/j.cie.2019.106035.

[11] P. Parikh, R. Meller, A travel-time model for a person-onboard order picking system, Eur. J. Oper. Res. 200 (2010) 385–394, https://doi.org/10.1016/j.ejor.2008.12.031.

[12] K. Roodbergen, I. Vis, A model for warehouse layout, IIE Trans. 8 (10) (2006) 799–811, https://doi.org/10.1080/07408170500494566.

[13] T. van Gils, A. Caris, K. Ramaekers, K. Braekers, Formulating and solving the integrated batching, routing, and picker scheduling problem in a real-life spare parts warehouse, Eur. J. Oper. Res. 277 (2019) 814–830, https://doi.org/10.1016/j.ejor.2019.03.012.

[14] R. de Koster, T. Le-Duc, N. Zaerpour, Determining the number of zones in a pick-and-sort order picking system, Int. J. Prod. Res. 50 (3) (2012) 757–771, https://doi.org/10.1080/00207543.2010.543941.

[15] T. Le-Duc, R. de Koster, Travel time estimation and order batching in a 2-block warehouse, Eur. J. Oper. Res. 176 (1) (2007) 374–388, https://doi.org/10.1016/j.ejor.2005.03.052.

[16] C. Petersen, Considerations in order picking zone configuration, Int. J. Oper. Prod. Manag. 22 (7) (2002) 793–805, https://doi.org/10.1108/01443570210433553.

[17] I. Van Nieuwenhuyse, R. de Koster, Evaluating order throughput time in 2-block warehouses with time window batching, Int. J. Prod. Econ. 121 (2) (2009) 654–664, https://doi.org/10.1016/j.ijpe.2009.01.013.

[18] E. Bottani, M. Cecconi, G. Vignali, R. Montanari, Optimisation of storage allocation in order picking operations through a genetic algorithm, Int. J. Logist. 15 (2) (2012) 127–146, https://doi.org/10.1007/978-3-319-17906-3_25.

[19] R. Manzini, R. Accorsi, M. Gamberi, S. Penazzi, Modeling class-based storage assignment over life cycle picking patterns, Int. J. Prod. Econ. 170 (2015) 790–800, https://doi.org/10.1016/j.ijpe.2015.06.026.

[20] G. Casella, A. Volpi, R. Montanari, L. Tebaldi, E. Bottani, Trends in order picking: a 2007–2022 review of the literature, Prod. Manuf. Res. 11 (1) (2023) 2191115, https://doi.org/10.1080/21693277.2023.2191115 article no.

[21] R. Manzini, M. Gamberi, A. Persona, A. Regattieri, Design of a class based storage picker to product order picking system, Int. J. Adv. Manuf. Technol. 32 (7–8) (2007) 811–821, https://doi.org/10.1007/s00170-005-0377-2.

[22] C. Theys, O. Bräysy, W. Dullaert, B. Raa, Using a TSP heuristic for routing order pickers in warehouses, Eur. J. Oper. Res. 200 (3) (2010) 755–763, https://doi.org/10.1016/j.ejor.2009.01.036.

[23] K.J. Roodbergen, G.P. Sharp, I.F.A. Vis, Designing the layout structure of manual order picking areas in warehouses, IIE Trans. 40 (11) (2008) 1032–1045, https://doi.org/10.1080/07408170802167639.

[24] T. van Gils, K. Ramaekers, K. Braekers, B. Depaire, A. Caris, Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions, Int. J. Prod. Econ. 197 (2018) 243–261, https://doi.org/10.1016/j.ijpe. 2017.11.021.

[25] H. Hwang, Y.H. Oh, C.N. Cha, A stock location rule for a low level picker-to-part system, Eng. Optim. 35 (3) (2003) 285–295, https://doi.org/10.1080/0305215031000136172.

[26] C. Petersen, An evaluation of order picking routeing policies, Int. J. Oper. Prod. Manag. 17 (11) (1997) 1098–1111, https://doi.org/10.1108/01443579710177860.

[27] G. Dukic, C. Oluic, Order-picking methods: improving order-picking efficiency, Int. J. Logist. Syst. Manag. 3 (4) (2007) 451–460, https://doi.org/10.1504/IJLSM.2007.013214.

[28] S.S. Rao, G.K. Adil, Optimal class boundaries, number of aisles, and pick list size for low-level order picking systems, IIE Trans. 45 (12) (2013) 1309–1321, https://doi.org/10.1080/0740817X.2013.772691.

[29] E. Bottani, R. Montanari, M. Rinaldi, Development and testing of software tool for warehouse design and picking optimisation, Int. J. Manag. Decis. Making 18 (2) (2019) 119–150, https://doi.org/10.1504/IJMDM.2019.098649.

[30] I. Manuj, J.T. Mentzer, M.R. Bowers, Improving the rigor of discrete-event simulation in logistics and supply chain research, Int. J. Phys. Distrib. Logist. Manag. 39 (3) (2009) 172–201, https://doi.org/10.1108/09600030910951692.

[31] R. Manzini, R. Accorsi, L. Pattitoni, A. Regattieri, A supporting decisions platform for the design and optimization of a storage industrial system, in: C. Jao (Ed.), Efficient Decision Support Systems - Practice and Challenges in Multidisciplinary Domains, IntechOpen, 2011, https://doi.org/10.5772/16861.

[32] R. Accorsi, R. Manzini, F. Maranesi, A decision-support system for the design and management of warehousing systems, Comput. Ind. 65 (1) (2014) 175–186, https://doi.org/10.1016/j.compind.2013.08.007.

[33] S.G. Ozden, A.E. Smith, K.R. Gue, A computational software system to design order picking warehouses, Comput. Oper. Res. 132 (105311) (2021), https://doi.org/10.1016/j.cor.2021.105311 article no.

[34] K.J. Roodbergen, Interactive Warehouse, a, 2019. Available at, http://www.roodbergen.com/warehouse/frames.htm.

[35] K.J. Roodbergen, Warehouse Optimizer By Kees Jan Roodbergen, b, 2019. Available at, http://www.roodbergen.com/whopt/.

[36] R. Montanari, E. Bottani, A. Volpi, F. Solari, N. Lysova, M. Bocelli, Warehouse Design and Management: a simulative approach to minimize the distance travelled by pickers, in: Proceedings of the 24thInternational Conference on Harbor, Maritime and Multimodal Logistics Modelling and Simulation, HMS 2022, 2022, https://doi.org/10.46354/i3m.2022.hms.005.

[37] C. Suppini, A. Volpi, F. Solari, E. Bottani, L. Tebaldi, M. Bocelli, R. Montanari, Performance analysis of traditional and trilateral forklifts in warehouse material handling: a technical-economic evaluation and simulation study, in: Proceedings of the 25th International Conference on Harbor, Maritime and Multimodal Logistic Modeling & Simulation(HMS 2023), 2023, https://doi.org/10.46354/i3m.2023.hms.001.

[38] R. Montanari, R. Micale, E. Bottani, A. Volpi, G. La Scalia, Evaluation of routing policies using an interval-valued TOPSIS approach for the allocation rules, Comput. Ind. Eng. 156 (2021), https://doi.org/10.1016/j.cie.2021.107256.

[39] H. Liu, F. Wang, J. Zhao, J. Yang, C. Tan, L. Zhou, Performance analysis of picking path strategies in Chevron layout warehouse, Mathematics (3) (2022) 10, https://doi.org/10.3390/math10030395.

Michele Bocelli is a Ph.D. Candidate in Industrial Engineering at the University of Parma. He graduated in Mechanical Engineering in 2019 at the University of Parma. Before applying for the Ph.D. program, he worked for three years as mechanical designer of bottling lines for mineral water for food use. His research field mainly concerns the optimization of production plants and logistics systems through innovative approaches. In addition, he expanded his field of research through parabolic flight campaigns, which allow to conduct experiments under conditions of altered gravity.

Eleonora Bottani is Full Professor of Industrial Logistics at the Department of Engineering for Industrial Systems and Technologies of the University of Parma since November 2019. She graduated (with distinction) in Industrial Engineering and Management in 2002 and got her Ph.D. in Industrial Engineering in 2006, both at the University of Parma. Her research activities concern logistics and supply chain management issues. She is author (or co-author) of >200 scientific papers (citations on Scopus>3600; H-index=31), referee for more than 60 international journals, editorial board member of five scientific journals, Associate Editor for various journals, and editor-in-chief of a scientific journal.

Andrea Volpi graduated cum laude in July 2003 in Mechanical Engineering at the University of Parma and then he started working in ICT field, RFID and IoT since January 2006

as Ph.D. student at the Industrial Engineering Department at the same University. He continued his studies as Lecturer and then as Associate Professor, focusing on research projects carried out in RFID Lab, a forefront laboratory in the same department. Technology is investigated as a driver for research activities mainly concerned with logistics and supply chain management issues; in fact, skills and competences developed are mainly related to RFID and logistics topics which are expressed in many papers produced.

Federico Solari is a Researcher and Lecturer at the Department of Engineering for Industrial Systems and Technologies of the University of Parma since September 2021. He graduated (with distinction) in Engineering for the Food Industry in 2008 and got his Ph.D. in Industrial Engineering in 2014, both at the University of Parma. His-research activities concern industrial plant logistics, industrial plant analysis and design, supply chain management, advanced industrial plant design also using simulative techniques, Design of Experiment and statistical analysis to develop, test and validate virtual models and digital twins of industrial systems. He authored (or co-authored) more than 30 publications indexed on Scopus and is one of the inventor of an Italian and international patent.

Natalya Lysova is Ph.D Candidate in Industrial Engineering at the University of Parma, in collaboration with FMB Eng. In. E. Srl. Her Ph.D. project is titled "Virtualisation approaches for industrial plants control and design". Her main research topics include simulation and optimization of industrial plants with the aid of CFD simulation, and simulation-based investigation and optimization of inventory management policies in the case of perishable products and/or system constraints. In addition, during her research activities, she has focused on the numerical simulation of ultraviolet treatments of solid and liquid foods.

Roberto Montanari is Full Professor of Industrial Plants at the Department of Engineering for Industrial Systems and Technologies, University of Parma. In addition, Prof. Montanari served as the Director of the Interdepartmental Centre for Packaging (CIPACK), and as the Delegate of the Rector for curricular and extracurricular internships. He is an esteemed member of the College of Professors in the Industrial Engineering Ph.D. program and actively participates in the internationalisation commission. Notably, he is responsible for overseeing the Double Degree program with NJIT (NJ - USA). He is author (or co-author) of >100 scientific papers (citations on Scopus>2000; H-index=23). Moreover, Prof. Montanari is the esteemed founder and partner of the academic spin-off FMB - Eng.In.E. SRL.