Journal of Machine Engineering, 2026, Vol. 26
ISSN 1895-7595 (Print) ISSN 2391-8071 (Online)

Received*: 16 February 2026* / Accepted: *21 April 2026* / Published online: *12 May 2026*

industrial robot,
pick and place,
sorting algorithm,
SPT

Kamil KROT[1*],
Bartosz POSKART[1],
Jakub MAZUR[1]

ALGORITHM SELECTION FOR DYNAMIC ROBOTIC PICK-AND-PLACE TASKS ON CONVEYOR BELTS UNDER TIME CONSTRAINTS

The use of robots in pick-and-place operations has been a well-known and widely used approach in production processes practically since the inception of robotic systems. Currently, however, increasingly stringent requirements are being placed on the control of robotic pick-and-place processes. This requires real-time robot control and the collection of objects of various classes in a random order and location within the robot's workspace. Collaboration between robots and vision systems monitoring objects appearing in front of the robot on a conveyor belt is becoming commonplace in industrial environments. This also necessitates the use of appropriate pick-and-place algorithms in robot control. This article presents an overview of component picking algorithms for robotic production lines. Key features of these algorithms are described. The effect of local object density for a selected object class on sorting performance at varying conveyor belt speeds was analysed. The study focused on the interaction between input flow characteristics and conveyor dynamics, assuming constant robot kinematic parameters and a fixed object placement point location. Furthermore, an approximate relationship describing the maximum allowable conveyor belt speed as a function of object density was derived, defining the stability limit of the SPT algorithm.

1. INTRODUCTION AND PROBLEM DEFINITION

Robotic sorting stations must decide in what order the robot should pick up detected objects to maximize work efficiency. The problem of determining the picking sequence is complex, corresponding to the problem of task scheduling. In practice, with a conveyor belt, each item has a limited time window for picking (before it leaves the robot's operating zone), which further complicates trajectory planning. Because an exact search of all permutations is computationally inefficient (or even impossible), the number of possible picking sequences for 15 items detected is: 15! = 1 307 674 368 000 permutations. Consequently, even with very small sets of items, a complete search of the solution space is already computationally

---

[1] Faculty of Mechanical Engineering, Wroclaw University of Science and Technology, Poland
[*] E-mail: kamil.krot@pwr.edu.pl
https://doi.org/10.36897/jme/220925

impractical (from several hours to several days of computation) or completely unacceptable in terms of time. Consequently, heuristic and greedy decision rules are commonly applied in real-time industrial systems, as they operate within a short decision horizon and provide low computational complexity.

The objective of this study is to analyse the performance limits and stability regions of selected greedy task sequencing algorithms (FIFO, SPT, modified NNF) in a dynamic conveyor-based pick-and-place system operating under time constraints, and to determine the parameter ranges in which each algorithm provides superior capture efficiency.

The central research problem addressed in this work is whether a single greedy sequencing rule is sufficient to ensure the best performance across the entire operating range of a dynamic conveyor-based sorting system.

The novelty of the described research results is the demonstration of the dependence of the efficiency of the greedy scheduling strategy on the object flow density and the conveyor belt speed, and the demonstration that no single algorithm provides dominant performance under all operating conditions.

2. RELATED WORK

Pick-and-place operations, in which a robot picks, carries, and then places an object in a desired location, are the quintessential robotic object manipulation problem. The use of robots in industry is becoming increasingly widespread, both in robotic machining and assembly operations, as well as in quality control. Improvements in the accuracy of gears used in machining robots are described in article [1]. The importance of maintaining the accuracy of industrial robots is emphasized in the paper [2], which develops a methodology and application for assessing the geometric and static accuracy of articulated industrial robots using the Extended-Double-Ball-Bar and Loaded-Double-Ball-Bar methods.

Robotic picking operation planning and robot trajectory design are widely discussed in the literature. In particular, the “Fast-Exploring Random Tree” (RRT) algorithms are extensively described. The paper [3] proposes a novel learning-based approach using multi-RRT (LM-RRT) for robot path planning in narrow passages. In the paper [4] propose Hybrid RRT, which divides the planning process into three parts: finding initial solutions by a dual-tree search, combining two trees into one, and optimizing the solution. In order to obtain an initial solution, Hybrid RRT implements a dual-tree search, which helps it find solutions faster than unidirectional searches. Article [5] proposes an improved artificial potential field and rapid expansion random tree (APF-RRT) hybrid algorithm for the mechanical arm path planning method. In [6], a new path planning algorithm using reinforcement learning based on a topological map was proposed. The proposed algorithm has a two-level structure.

In the following discussion, we selected scheduling algorithms for retrieving objects from a conveyor belt. They were characterized formally and mathematically to enable unambiguous implementation and testing. Greedy algorithms [7] are a class of decision-making algorithms in which a solution is built iteratively by selecting the locally best option at each step according to an adopted cost criterion, without considering the impact of this decision on subsequent steps of the process. This approach does not assume the existence of a planning horizon or the analysis of alternative decision sequences, resulting in low computational complexity and implementation simplicity. However, it does not guarantee achieving a globally optimal solution, particularly in sequential and dynamic problems.

The simplest approach is FIFO (First In, First Out) – the robot retrieves objects in the order in which they were detected, meaning the first item found is processed first. This strategy does not require time estimation or complex trajectory calculations, making it computationally light. The main advantage of this scheduling algorithm is its constant complexity. In particular, it does not require the calculation and sorting of queue capture times, which translates into a very low computational cost. Moreover, under underload conditions, it is optimal for many classes of scheduling problems, where the problem of scheduling tasks in real-time systems is analysed [8]. However, this algorithm proves inefficient under overload conditions.

Shortest Processing Time (SPT) is a scheduling algorithm in robotics' task queuing theory, which involves always picking the item that can be picked in the shortest time [9]. SPT is a simple scheduling rule that prioritizes tasks by first selecting the one requiring the shortest processing time, then the next, the shortest, and so on, to minimize the total waiting time, the lead time, and the work-in-progress inventory. This approach is effective because it allows for the rapid completion of short tasks, but longer tasks can be significantly delayed and sometimes skipped.

The application of the Nearest Neighbour First (NNF) algorithm in an object retrieval application with a robot end effector involves calculating the distance between all object retrieval points and the robot's current TCP position. Priority “1” (retrieved first) is assigned to the element whose distance from the TCP is the smallest [10]. The NNF algorithm is sometimes used to solve the traveling salesman problem, but in a pick-and-place application, the deposit point of a retrieved object of a given class becomes the new TCP starting position, requiring a recalculation of the distance between the TCP and all object retrieval points currently submitted for retrieval [11].

The difference between the Shortest Processing Time (SPT) algorithm and the Nearest Neighbour (NNF) algorithm is conceptual, although in simplified implementations, both approaches can lead to similar decisions. The Nearest Neighbour algorithm relies solely on geometric criteria, selecting the object requiring the shortest trajectory, regardless of subsequent operation steps. In contrast, the SPT algorithm relies on estimating the total handling time, which takes into account speed and acceleration constraints, and–in a more advanced approach–the full dynamics of the robot, which can lead to a different ordering of retrieved objects than distance-based heuristics.

Optimization of the pick-and-place sequence on the conveyor cannot be effectively solved with static algorithms (FIFO, SPT) if the system load exceeds the manipulator's capacity - the problem loses the character of a classical queuing problem and becomes a problem of dynamic planning of a sequence of decisions in real time. Sequential planning algorithms with a decision horizon are a class of optimization methods in which the current decision is selected based on an analysis of the consequences of an entire sequence of future actions within a finite planning horizon. Unlike greedy algorithms, these methods do not minimize the cost of a single step but rather strive to optimize the total cost of the sequence, taking into account time dependencies, system dynamics, and the impact of the current decision on the availability of subsequent decision options. This allows them to achieve globally optimal or near-optimal solutions in sequential and dynamic problems, at the cost of increased computational complexity.

The exhaustive search algorithm [12] systematically generates and evaluates all feasible solutions within the considered solution space. The algorithm begins by defining a set of all possible decision configurations, which, depending on the nature of the problem, can be structured as a set of combinations, permutations, or decision vectors. Each feasible solution is then independently evaluated by calculating the objective function value, which describes the quality of the given solution according to the adopted optimization criterion. During the search, the algorithm stores information about the best solution found so far, updating it each time a configuration encounters a lower (in the case of minimization) or higher (in the case of maximization) objective function value. After completing the search of the entire solution space, the algorithm returns a solution with the optimal objective function value, guaranteeing a globally optimal solution. The advantage of this method is its conceptual simplicity and full guarantee of optimality, but its fundamental limitation remains its very high computational complexity, which increases exponentially or more rapidly with the number of decision variables, which in practice limits its application to small-scale problems.

Recent survey studies indicate that task scheduling in mobile robotic systems is dominated by heuristic and hybrid strategies rather than full decision-horizon optimization due to scalability and real-time constraints [13]. In particular, constructive heuristics have been shown to provide efficient schedules for transport and pick robots in logistics applications under practical computation limits [14], while domain-specific heuristic developments further demonstrate performance improvements in multi-robot pick-and-place systems compared to traditional rule-based methods [15]. Although learning-based and hybrid approaches are emerging, they still face challenges in balancing adaptability and tractability for on-line deployment.

3. METHODOLOGY

When planning research on the evaluation of object picking algorithms in dynamic Pick-and-Place operations, it was necessary to define criteria for evaluating sorting algorithms, which were then subjected to comparative testing. Based on the state-of-the-art review, it was assumed that before implementing the algorithms in a simulation environment, the feasibility of their further development should be verified at the computational stage. The primary criterion for further algorithm implementation is the computational time required to determine the order of item picks. This criterion stems directly from the requirements of the robotic system, in which the order of picks must be determined before the robot is ready to perform the operation. Therefore, the scheduling algorithm must not constitute a bottleneck at the workstation or cause delays in the operational cycle.

The experiments evaluated three greedy scheduling algorithms commonly applied in robotic sorting and manipulation tasks:

- First-In–First-Out (FIFO),
- Shortest Processing Time (SPT),
- a modified Nearest Neighbour First (NNF) algorithm.

The primary objective of the first experiment is to analyse the impact of local object density for a selected object class on sorting performance under varying conveyor belt speeds. The study focuses on the interaction between input stream characteristics and transport dynamics, while assuming constant robot kinematic parameters and a fixed location of the object placement point. The main performance criterion considered in this stage of the research is the computational execution time of the algorithms and, indirectly, their computational complexity. Only algorithms that satisfy the real-time computational constraints of the robotic system are qualified for further functional evaluation, including their influence on overall system throughput and operational efficiency. As an initial step, computational execution time tests were conducted for problem instances with cardinality

N={1, 5, 10, 15, 20} objects present simultaneously within the robot operating zone. The results of these tests provide an initial assessment of algorithm scalability and enable the identification of solutions suitable for real-time robotic applications.

3.1. SIMULATION ENVIRONMENT

The simulation environment was developed in-house in Python using the Pygame library, which ensured full control over the model structure, enabled the implementation of custom control algorithms and decision logic, and provided straightforward process visualization. Robot motion was modeled as kinematically constrained motion (with velocity and acceleration limits), represented by either a triangular or trapezoidal velocity profile, depending on whether the maximum velocity vmax could be reached along a given trajectory. The conditions for switching between motion profiles, as well as the total trajectory execution time, were determined analytically based on classical kinematic equations for motion with bounded velocity and acceleration. The condition for triangular or trapezoidal motion profiles is defined as follows (1):

$$d_{min} = \frac{{V_{max}}^2}{a_{max}} \tag{1}$$

If $d_{total} \geq d_{min}$, a trapezoidal velocity profile is applied. If $d_{total} < d_{min}$, a triangular velocity profile is used.
where:
d_{total} - total travel distance of the end-effector,
d_{min} – minimum distance required to reach V_{max},
v_{max} – declared maximum linear velocity of the robot,
a_{max} – declared maximum acceleration of the robot.

Total motion time for a trapezoidal velocity profile is defined as follows (2):

$$t_{total} = \frac{d_{total}}{v_{max}} + \frac{v_{max}}{a_{max}} \tag{2}$$

Total motion time for a triangular velocity profile is defined as follows (3):

$$t_{total} = 2\sqrt{\frac{d_{total}}{a_{max}}} \tag{3}$$

In the simulation, kinematic parameters typical for delta robots were adopted, based on reference data for the ABB IRB 360-1/1130 robot, i.e., V_{max}=10 m/s and a_{max}=100 m/s^2. The detailed scope of the adopted simulation constraints is presented in Section 3.3.

The developed simulation environment enables the comparison of task sequencing algorithms based on defined performance metrics (% capture rate, OEE, and PPM). The capture rate is defined as the ratio of successfully picked objects to the total number of objects available in the system under given process parameters (robot velocity, acceleration, conveyor speed, and object density per unit length). The OEE indicator is defined as OEE = Availability· Performance·Quality with the assumptions Availability = 1 (no failures) and Quality=1 (100% picking success rate), which reduces it to the performance component. The Performance term is calculated as the ratio of the robot's active motion time to the total system operation time, assuming the execution of the shortest trajectories. Additionally, the PPM (parts per minute) metric is used, defined as the number of complete pick-and-place cycles executed per unit time (one minute).

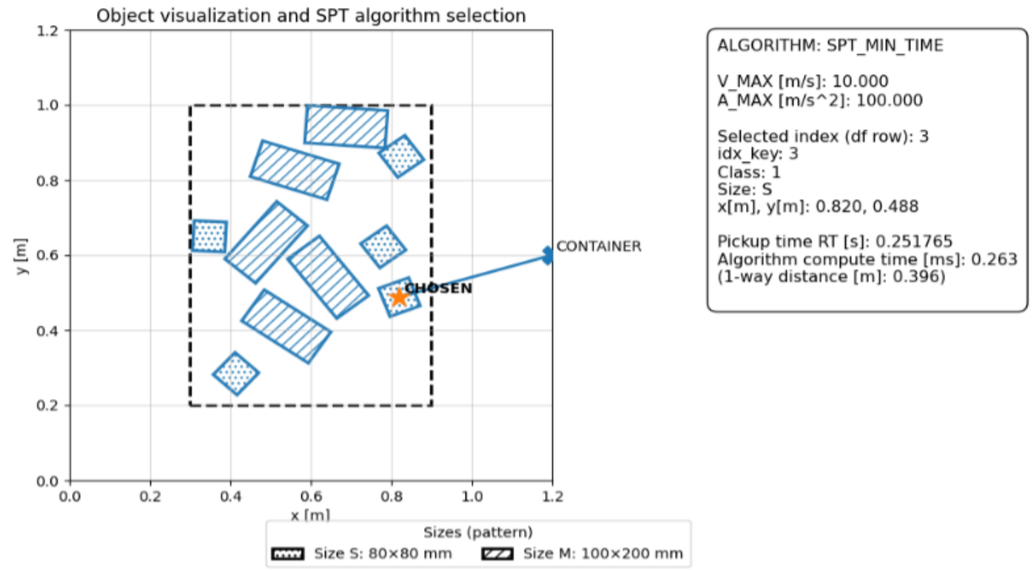
3.2. ALGORITHMS COMPUTATION TIME

The FIFO algorithm belongs to the class of greedy algorithms with a simple decision structure and low computational complexity. In its current form, it selects the next element to be retrieved based on the minimum Y coordinate value, interpreted as the element's position along the conveyor, i.e., its distance from the service boundary line. Simulation results showed that the time to determine the pick order remains below 1 ms, even for the case of analysing 20 items. The resulting computation time does not limit the station's work cycle and allows for timely decision transfer to the robot control system

The SPT algorithm, similarly to the FIFO algorithm, belongs to a class of greedy algorithms with a simple decision structure and low computational complexity. SPT suggests selecting the next item to be picked based on the shortest expected operation completion time, defined as the time required for the robot to complete the item pick cycle. The simulation resulted in a pick sequence determination time of less than 1 ms, even for the case of analysing 20 elements. The resulting computation time does not limit the station's work cycle and enables timely transfer of decisions to the robot control system

The simulation results demonstrate that the SPT algorithm, as a distance-driven greedy optimization strategy, exhibits an inherent selection bias toward elements located closer to the placement point, resulting in a systematically asymmetric utilization of the feeder space, as shown in Fig. 1.

The NNF algorithm is conceptually similar to the SPT algorithm, as it relies on the greedy selection of an element that minimizes the cost of a single pick operation. In its classical form, this algorithm favours elements located closest to the storage bin (storage point). Therefore, this paper proposes a modified version of the NNF algorithm, in which the element selection criterion is the distance from the centre of the belt conveyor. This allows for picking elements from the centre of the conveyor (without favouring only one side) while maintaining a simple decision structure. The simulation results showed that the modified NNF algorithm remains a greedy algorithm with low computational complexity, ensuring that its execution time meets the requirements of the robotic system and does not limit the workflow of the station. Even for the case of analysing 20 elements, the computation time remains below 1 ms (Table 1).



```
ALGORITHM: SPT_MIN_TIME

V_MAX [m/s]: 10.000
A_MAX [m/s^2]: 100.000

Selected index (df row): 3
idx_key: 3
Class: 1
Size: S
x[m], y[m]: 0.820, 0.488

Pickup time RT [s]: 0.251765
Algorithm compute time [ms]: 0.263
(1-way distance [m]: 0.396)
```

Fig. 1. Object on conveyor visualization for simulation object picking using the SPT algorithm

Table. 1 Comparison of the average computation times of the FIFO, SPT, and modified NNF algorithms for different numbers of objects on the conveyor (mean values obtained from 10 measurements)

Number of objects on conveyor	1	5	10	15	20
Computational time FIFO [ms]	0.091	0.146	0.206	0.299	0.355
Computational time SPT [ms]	0.123	0.216	0.222	0.296	0.356
Computational time modified NNF [ms]	0.101	0.175	0.238	0.323	0.382

Simulation studies conducted for the numbers of objects on the conveyor: 1, 5, 10, 15, and 20 compared FIFO, SPT, and modified NNF algorithms show that the computation time is less than 1 ms in all cases. Therefore, this does not constitute a limitation for the sorting operation.

3.3. ALGORITHMS CAPTURE EFFICIENCY

Simulation studies of object picking and placing algorithms were conducted under a set of controlled assumptions reflecting an industrial conveyor-based sorting scenario. The analysed process involves an industrial delta robot performing pick-and-place operations on objects transported by a conveyor belt.

The simulation environment was defined as follows:

- the conveyor belt section considered for the analysis corresponds to the effective robot operating zone and has dimensions of 600 mm in width and 800 mm in length,
- the conveyor belt linear speed is adjustable within the range of 0.1–1.0 m/s,
- the industrial delta robot is characterised by a maximum linear velocity of 10 m/s and a maximum acceleration of 100 m/s^2,
- the robot workspace is defined as a square area of 1200 × 1200 mm,
- the object picking time is fixed at 0.1 s,
- the object placing time is fixed at 0.1 s,
- each simulation scenario consists of 5000 objects transported through the system,
- the object density along the conveyor belt is varied in the range of 1–20 objects per metre of conveyor length,
- objects are distributed with uniform spacing along the conveyor axis, while their lateral positions are randomly generated,
- identical object sets are used for testing all algorithms to ensure comparability of results.

To isolate the influence of the picking order strategy, several simplifying assumptions were adopted. In particular, a 100% object picking success rate was assumed, regardless of object geometry, material properties, mass, or orientation. Limitations related to the mechanical design of the end effector, grasp feasibility, and actuator activation delays were not considered. These simplifications allow the study to focus exclusively on the algorithmic aspects of task sequencing. A simulation model that takes into account the actual picking efficiency would affect the calculation of algorithmic efficiency across the entire spectrum of the studied object density range and for all conveyor speeds, and would make it impossible to achieve 100% picking efficiency.

Based on the above-defined assumptions, a series of simulation experiments was conducted to evaluate the performance of different object picking and placing algorithms under varying conveyor operating conditions. The experiments systematically varied the object density on the conveyor belt as well as the conveyor linear speed, while maintaining all other parameters constant. Three task sequencing strategies were analysed - FIFO, SPT and modified NNF algorithm. To ensure a fair comparison, identical object distributions were used across all algorithmic variants for each experimental configuration.

As a result of testing the picking algorithms with the adopted model simplifications, the loss of the ability to capture 100% of objects follows the increasing conveyor belt speed and the increasing object density on the conveyor. This is due to the inability to complete all the elementary robot movements required to pick up “x” objects, in a time period determined by the conveyor speed. The results of a representative experiment obtained for the FIFO algorithm are presented in Table 2. The conducted simulation experiments enabled a comparative evaluation of the FIFO, SPT, and modified NNF algorithms in terms of object capture efficiency, overall equipment effectiveness (OEE), and picks per minute (PPM). The results clearly indicate that the SPT algorithm is the dominant solution, achieving the highest capture efficiency for the vast majority of combinations of object density and conveyor belt speed.

The FIFO and modified NNF algorithms outperform the remaining methods only within narrow, localized regions of the parameter space, which suggests that their advantages are situational rather than global. At the same time, the results demonstrate that maximizing capture efficiency is not always appropriate objective for system operation. Allowing a controlled reduction in capture efficiency, for example to a level of 90%, enables an increase in object density or conveyor speed, which directly translates into higher system throughput expressed by PPM and a potential improvement in OEE.

Table 2. Percentage capture efficiency of the FIFO algorithm depending on conveyor linear speed and object density on the conveyor

Object density [objects/m]	**Conveyor belt speed [m/s]**						
	0.1	0.2	0.3	0.4	0.6	0.8	1,0
1	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**
2	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**	92,8	74,1
3	**100,0**	**100,0**	**100,0**	**100,0**	83,1	62,4	50,1
4	**100,0**	**100,0**	**100,0**	93,6	62,5	47,0	37,7
5	**100,0**	**100,0**	99,6	74,7	50,0	37,5	30,2
6	**100,0**	**100,0**	82,7	62,2	41,6	31,4	25,2
7	**100,0**	**100,0**	71,0	53,4	35,8	27,0	21,7
8	**100,0**	92,7	62,0	46,7	31,4	23,7	19,0
9	**100,0**	82,2	55,0	41,5	27,7	21,0	16,9
10	**100,0**	74,0	49,6	37,3	25,1	19,1	15,3
11	**100,0**	67,2	45,1	33,9	22,8	17,2	14,0
12	**100,0**	61,8	41,3	31,2	20,9	15,7	12,8
13	**100,0**	56,9	38,3	28,7	19,4	14,6	11,9
14	**100,0**	52,7	35,4	26,6	18,0	13,7	11,0
15	98,5	49,4	33,1	24,9	16,8	12,8	10,3
16	92,0	46,4	31,1	23,4	15,8	12,0	9,6
17	86,6	43,6	29,2	22,0	14,9	11,3	9,1
18	81,9	41,2	27,7	20,8	14,1	10,7	8,6
19	77,6	39,1	26,2	19,7	13,3	10,1	8,2
20	73,6	37,0	24,9	18,8	12,7	9,6	7,8

The presented algorithm selection matrix (Table 3) facilitates the identification of operating parameter ranges in which individual algorithms exhibit superior performance and provides a practical basis for selecting an appropriate sorting strategy depending on process requirements.

Based on the simulation results, the capture efficiency values of the SPT algorithm (Table 4) were tabulated as a function of object density and conveyor belt speed. The SPT algorithm was selected for detailed presentation, as it achieves the highest capture efficiency for the majority of operating conditions, in accordance with the algorithm selection matrix presented in Table 3.

The analysis of boundary points corresponding to operating conditions close to 100% capture efficiency enabled the derivation of an approximate relationship describing the maximum allowable conveyor belt speed as a function of the current object density on the belt, expressed as:

$$v = \frac{1.243}{d + 0.228}\ [\mathrm{m/s}] \tag{3}$$

where:

v - linear speed of the conveyor belt [m/s],

d - number of objects per metre of conveyor length [objects/m].

The relationship between conveyor speed and object density was determined empirically based on simulation results obtained for the SPT algorithm. For each density level, the maximum conveyor speed ensuring 100% capture efficiency was identified. When the threshold lay between two sampled speed values, it was estimated using linear interpolation. The resulting set of boundary points was approximated using a hyperbolic function. This expression should be interpreted as an empirical approximation of the system's operating boundary for the analysed configuration, rather than a universal physical relationship.

The fitted model achieved a coefficient of determination of $R^2 = 0.986$ with a mean absolute error of $MAE = 0.021$m/s and a root mean square error of $RMSE = 0.029$m/s, indicating a very good agreement with the observed trend within the investigated range.

Table 3. Algorithm selection matrix based on object density and conveyor belt speed

Object density [objects/m]	Conveyor belt speed [m/s]						
	0.1	0.2	0.3	0.4	0.6	0.8	1
1	SPT	SPT	SPT	SPT	SPT	SPT	SPT
2	SPT	SPT	SPT	SPT	SPT	FIFO	SPT
3	SPT	SPT	SPT	SPT	FIFO	SPT	SPT
4	SPT	SPT	SPT	FIFO	SPT	SPT	SPT
5	SPT	SPT	FIFO	FIFO	SPT	SPT	SPT
6	SPT	SPT	**NNF(mod)**	SPT	SPT	SPT	SPT
7	SPT	FIFO	**NNF(mod)**	SPT	SPT	SPT	SPT
8	SPT	FIFO	SPT	SPT	SPT	SPT	SPT
9	SPT	FIFO	SPT	SPT	SPT	SPT	SPT
10	SPT	**NNF(mod)**	SPT	SPT	SPT	SPT	SPT
11	SPT	**NNF(mod)**	SPT	SPT	SPT	SPT	SPT
12	SPT	SPT	SPT	SPT	SPT	SPT	SPT
13	SPT	SPT	SPT	SPT	SPT	SPT	SPT
14	FIFO	SPT	SPT	SPT	SPT	SPT	SPT
15	FIFO	SPT	SPT	SPT	SPT	SPT	SPT
16	FIFO	SPT	SPT	SPT	SPT	SPT	SPT
17	FIFO	SPT	SPT	SPT	SPT	SPT	SPT
18	FIFO	SPT	SPT	SPT	SPT	SPT	SPT
19	FIFO	SPT	SPT	SPT	SPT	SPT	SPT
20	**NNF(mod)**	SPT	SPT	SPT	SPT	SPT	SPT

The obtained model defines the stability boundary of the SPT algorithm and can be applied for adaptive control of the conveyor belt speed in systems equipped with vision-based estimation of object stream density. This approach allows maximizing the number of picks while maintaining a high level of sorting efficiency.

Table 4. Percentage capture efficiency of the SPT algorithm depending on conveyor linear speed and object density on the belt

Object density [objects/m]	Conveyor belt speed [m/s]						
	0.1	0.2	0.3	0.4	0.6	0.8	1
1	**100**	**100**	**100**	**100**	**100**	**100**	**100**
2	**100**	**100**	**100**	**100**	**100**	92,2	74,44
3	**100**	**100**	**100**	**100**	82,28	63,42	51,12
4	**100**	**100**	**100**	90,44	63,52	48,66	39,15
5	**100**	**100**	94,84	74,18	51,76	39,49	31,69
6	**100**	**100**	81	63,03	43,68	33,26	26,64
7	**100**	99,44	70,89	55,05	38,02	28,84	23,09
8	**100**	89,16	62,86	48,71	33,5	25,38	20,35
9	**100**	80,36	56,49	43,72	30,02	22,68	18,17
10	**100**	73,32	51,5	39,79	27,19	20,53	16,41
11	**100**	67,38	47,18	36,37	24,82	18,75	14,98
12	**100**	62,61	43,7	33,59	22,82	17,21	13,73
13	**100**	58,19	40,63	31,2	21,24	16	12,76
14	98,9	54,51	37,9	29,06	19,78	14,89	11,9
15	93,48	51,34	35,63	27,29	18,49	13,92	11,12
16	88,34	48,4	33,63	25,72	17,41	13,09	10,46
17	83,82	45,82	31,79	24,28	16,41	12,35	9,84
18	79,84	43,65	30,16	23,04	15,56	11,7	9,33
19	76,12	41,49	28,63	21,86	14,77	11,08	8,85
20	72,82	39,54	27,3	20,85	14,05	10,57	8,44

2.CONCLUSION

The conducted study confirms that although the analysed FIFO, SPT, and NNF algorithms belong to the class of greedy algorithms, their simple decision structure and very low computational cost make them particularly suitable for real-time pick-and-place sorting systems operating under strict time constraints. In industrial practice, such algorithms are widely applied precisely because they enable rapid decision making without introducing delays into the robot control cycle.

At the same time, the simulation results clearly demonstrate that the use of a single sorting algorithm across the entire range of operating conditions does not lead to the best picking performance. The effectiveness of individual algorithms strongly depends on process parameters, in particular on the relationship between object density on the conveyor belt and its linear speed. For this reason, this work proposes an approach based on selecting the sorting strategy dynamically, depending on the current operating conditions of the system.

A key outcome of the research is the algorithm selection matrix, which enables the unambiguous identification of parameter ranges in which a given algorithm provides the highest capture efficiency. The matrix constitutes a practical decision-support tool, allowing adaptive switching between sorting strategies in order to maximize the utilization of a single robot, without the need to increase the number of manipulators in the system.

Additionally, the derived approximate relationship describing the maximum allowable conveyor belt speed as a function of object density defines the stability boundary of the SPT algorithm. This model can be directly applied to adaptive conveyor speed control, particularly in systems equipped with vision-based estimation of object stream density. Such an approach enables dynamic adjustment of transport parameters to ensure reliable handling of all objects while maximizing system throughput using only a single robot.

Future work should focus on evaluating more advanced task sequencing approaches, in particular sequential planning algorithms with a full decision horizon, which inherently require longer computation times and may not always yield globally most efficient solutions due to the combinatorial growth of the solution space. A systematic comparison of their performance against greedy algorithms would allow for a more comprehensive assessment of the trade-offs between solution quality, computational cost, and practical applicability in real-time robotic pick-and-place systems.

REFERENCES

[1] YAQOOB M.F., GOLLEE C., SCHENKE C., IHLENFELDT S., 2025, *Concept of Developing a Gear Selection Tool for Improved Accuracy in Industrial Robotics*, Journal of Machine Engineering, 25/2,138–148, https://doi.org/10.36897/jme/204359.

[2] MARWITZ J.A., THEISSEN N.A., GONZALEZ M.K.B., et al., 2022, *Accuracy Assessment of Articulated Industrial Robots Using the Extended- and the Loaded-Double-Ball-Bar*, Journal of Machine Engineering, 22/2, 80–98, https://doi.org/10.36897/jme/149413.

[3] WANG W., ZUO L., XU X., 2018, *A Learning-Based Multi-RRT Approach for Robot Path Planning in Narrow Passages*, J. Intell. Robot. Syst., 90, 81–100.

[4] MASHAYEKHI R., IDRIS M.Y.I., ANISI M.H., AHMEDY I., 2020, *Hybrid RRT: A Semi-Dual-Tree RRT-Based Motion Planner*, IEEE Access, 8, 18658–18668.

[5] YUAN Q., YI J., SUN R., BAI H., 2021, *Path Planning of a Mechanical Arm Based on an Improved Artificial Potential Field and a Rapid Expansion Random Tree Hybrid Algorithm*, Algorithms 14, 321.

[6] WEN S., JIANG Y., CUI B., GAO K., WANG F.A., 2022, *Hierarchical Path Planning Approach with Multi-SARSA Based on Topological Map*, Sensors, 22, 2367.

[7] TAMALI A., AMARDJIA N., TAMALI M., 2024, *Distributed and Autonomous Multi-Robot for Task Allocation and Collaboration Using a Greedy Algorithm and Robot Operating System Platform*, IAES International Journal of Robotics and Automation (IJRA), 13/2, 205.

[8] HUANG Y., CHIBA R., ARAI T., UEYAMA T., OTA J., 2015, *Robust Multi-Robot Coordination in Pick-And-Place Tasks Based on Part-Dispatching Rules*, Robotics and Autonomous Systems, 64, 70–83, https://doi.org/10.1016/j.robot.2014.10.018.

[9] WANG H., CHEN W., 2021, *Task Scheduling for Transport and Pick Robots in Logistics: a Comparative Study on Constructive Heuristics*, Auton. Intell. Syst. 1, 17, https://doi.org/10.1007/s43684-021-00017-9.

[10] ARANA G.D., JOERGER M., SPENKO M., 2018, *Local Nearest Neighbour Integrity Risk Evaluation for Robot Navigation*, IEEE Int. Conf. Robot. Autom., 2328–2333.

[11] KUMAR R., KUMAR S., LAL S., CHAND P., 2014, *Object Detection and Recognition for a Pick and Place Robot*, Computer Science, Engineering, Asia – Pacific World Congress on Computer Science and Engineering, https://doi.org/10.1109/APWCCSE.2014.7053853, 2014.

[12] ZHOU M., JIANG R., 2022, *Optimal Strategy for Pick-And-Place System with Two Robots*, J. Physics Conf. Ser., 2216, 012022.

[13] REMA C., COSTA P., SILVA M., PIRES E., 2025, *Task Scheduling with Mobile Robots - a Systematic Literature Review*, Robotics, 14, 75, https://doi.org/10.3390/robotics14060075.

[14] WANG H., CHEN W., 2021, *Task Scheduling for Transport and Pick Robots in Logistics: A Comparative Study on Constructive Heuristics*, Autonomous Intelligent Systems, 1, https://doi.org/10.1007/s43684-021-00017-9.

[15] TECK S., VANSTEENWEGEN P., DEWIL R., 2023, *An Efficient Multi-Agent Approach to Order Picking and Robot Scheduling in a Robotic Mobile Fulfilment System*, Simul. Model. Pract. Theory, 127, 102789, https://doi.org/10.1016/j.simpat.2023.102789.