

Article

RaapWaste: Robot- and Application-Agnostic Planning for Efficient Construction and Demolition Waste Sorting

Konstantinos Kokkalis ^{1,2,*}, Fotios K. Konstantinidis ¹ , Maria Koskinopoulou ^{1,3} , Georgios Tsimiklis ¹, Angelos Amditis ¹  and Panayiotis Frangos ²

¹ Institute of Communication and Computer Systems (ICCS), National Technical University of Athens, 9 Iroon. Polytechniou Str., Zografou, 15773 Athens, Greece; fotios.konstantinidis@iccs.gr (F.K.K.); m.koskinopoulou@hw.ac.uk (M.K.); georgios.tsimiklis@iccs.gr (G.T.)

² School of Electrical and Computing Engineering, National Technical University of Athens, 9 Iroon. Polytechniou Str., Zografou, 15773 Athens, Greece; pfrangos@central.ntua.gr

³ Institute of Signals, Sensors and Systems (ISSS), School of Engineering & Physical Sciences (EPS), Heriot-Watt University, Edinburgh EH141NF, UK

* Correspondence: konstantinos.kokkalis@iccs.gr

Abstract

Robotic waste sorting systems offer a scalable and consistent alternative to manual sorting for Construction and Demolition Waste (CDW) by reducing labor-intensive tasks and exposure to hazardous conditions, while enabling the extraction of high-purity materials (e.g., polymers) from the waste streams. Despite advancements in perception systems, manipulation and planning remain significant bottlenecks, limiting widespread adoption due to high complexity and cost. This paper introduces RaapWaste, a robot- and application-agnostic planning framework specifically designed for waste sorting, addressing challenges in motion planning, scheduling, and real-world integration. Built on open-source resources, RaapWaste employs a modular and flexible architecture, enabling integration of diverse planning techniques and scheduling strategies. The framework aims to simulate the performance of real-world sorting equipment (e.g., robots, grippers). To evaluate its effectiveness, we conducted simulations with articulated and delta robots, as well as real-world tests on CDW sorting. Metrics such as the Sorting Throughput (ST) and Sorting Ratio (SR) reveal the RaapWaste's capability across different waste sorting cases. In simulation, the delta robot achieved an SR exceeding 95%, while the UR5e showed consistent performance. In real-world CDW experiments, the system achieved a peak SR of 99% and maintained 80% using the SPT scheduler.

Keywords: CDW sorting; robot sorting; industrial automation; RaapWaste



Academic Editors: Michał Juszczyk, Tomáš Hanák and Michał Pyzalski

Received: 29 June 2025

Revised: 28 July 2025

Accepted: 4 August 2025

Published: 12 August 2025

Citation: Kokkalis, K.; Konstantinidis, F.K.; Koskinopoulou, M.; Tsimiklis, G.; Amditis, A.; Frangos, P. RaapWaste: Robot- and Application-Agnostic Planning for Efficient Construction and Demolition Waste Sorting. *Sustainability* **2025**, *17*, 7293. <https://doi.org/10.3390/su17167293>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

CDW is responsible for more than 30% of waste throughout the world, resulting in the generation of approximately 0.7 billion tonnes per year [1]. It is estimated that, on average, 35% of CDW is landfilled, with 75% having potential added value, resulting in missed opportunities for material recovery and increased pressure on landfills. Furthermore, CDW can include highly hazardous materials like asbestos, lead, mercury and coal ash, which should not migrate to the soil, but can be a valuable source for valorization, e.g., recycled cement, concrete, and ceramics [2,3].

CDW sorting is among the most demanding real-world sorting tasks due to the heterogeneous nature of the materials involved. The materials are highly irregular, varying

in shape, size, and density, which makes automated handling difficult. Conventional automated sorting systems rely on a specific physical property of the materials, which CDW often lacks, and generally do not possess the adaptability and versatility to sort CDW efficiently. Robotic sorting, on the other hand, does not depend on specific item properties, can adapt to changing conditions, and integrates seamlessly into various production lines.

Compared to manual sorting, robots provide a consistent, cost-effective, and scalable alternative, enabling human resources to be allocated to less repetitive, exhausting, and hazardous tasks. Recent advances in artificial intelligence and sensor technologies have significantly improved the vision capabilities of these systems, reaching performances of more than 95% for detection and classification accuracy of waste [4], making robotic sorting a more viable solution.

Despite their merits, robotic systems still face key adoption barriers due to lower throughput and high initial costs compared to traditional sorting technologies. Improving robotic sorting performance depends on optimizing sorting accuracy and rate in order to enhance picking success and cycle times. In addition, simulation and testing of planning parameters is critical, but constrained by the specificity of existing robot simulators, which often limit sensing hardware compatibility.

State-of-the-art solutions address sorting challenges by tailoring systems to specific applications and input materials, often at the expense of flexibility and versatility [5]. Generic solutions need to manage waste variability in manipulation, decision-making, and sensing without compromising performance. The recent literature on robotic sorting systems has largely focused on perception, where classification accuracy is most commonly above 90% for a non-trivial perception use case [6].

Nevertheless, manipulation and planning capabilities have not kept pace with sensing technologies. Most existing research on planning addresses sorting from narrow perspectives, focusing on specific applications, robot models, or simple decision rules. Motion planning techniques are often tailored to specific robots, limiting their adaptability. For example, methods designed for delta robots are unsuitable for articulated robots, which may be preferred for their versatility and workspace reach.

The development of robotic sorting systems is directly aligned with the objectives of the European Green Deal and the EU Circular Economy Action Plan, which call for efficient resource recovery, waste reduction, and sustainable reuse of construction and demolition materials. By improving the precision and adaptability of sorting operations in heterogeneous waste streams, frameworks like *RaapWaste* contribute to reducing landfilling, lowering greenhouse gas emissions associated with virgin material extraction, and increasing the quality of secondary raw materials available for valorization. This positions robotic sorting not just as a technical challenge, but as a key enabler for sustainable industrial transformation.

To address the aforementioned challenges, this work introduces *RaapWaste*, a modular and robot-agnostic planning framework designed to optimize robotic material sorting through the integration of task-level scheduling and motion-level planning. The system architecture separates high-level decision-making from low-level trajectory generation, enabling the use of heterogeneous manipulators and end-effectors within a unified planning pipeline. The framework is then tested with a real-world setup for a challenging application, CDW sorting. The contributions of our work are summarized as follows:

- *RaapWaste*: A robot- and application-agnostic planning framework that decouples task scheduling from motion planning for flexible deployment across waste sorting platforms.

- The integration of open-source frameworks ROS2 and MoveIt2 to ensure transparency, modularity, and adaptability, while enabling seamless incorporation of novel planning, kinematics, and control algorithms.
- The creation of a generic simulation environment that can be used for initial testing, debugging, and benchmarking before proceeding to the real system operation.
- A flexible architecture supporting First-In First-Out (FIFO) and SPT scheduling, projection-based grasp evaluation, and separate picking and placing pipelines for optimized trajectories.
- Comprehensive validation in simulation and real-world setup, using Universal Robots UR5e and FANUC M-3iA/6S robots.
- Evaluation with a real-world dataset from CDW management facilities

The remainder of this paper is structured as follows. Section 2 reviews related work in robotic waste sorting. Section 3 introduces the system architecture, while Section 4 details the implementation of the RaapWaste framework. Section 5 presents experimental results from both simulation and real-world setups. Section 6 discusses key findings, and Section 7 concludes the paper with future directions.

2. Related Work

This section reviews prior work on robotic waste sorting, especially CDW sorting, analyzing the selection of robotic configurations, end-effectors, motion planning methodologies, and scheduling strategies to contextualize the need for a more adaptable and versatile planning framework. In Table 1, some important works on robotic waste sorting have been gathered. For each paper, we record the robot, the gripper, the waste material to be sorted and whether experiments with simulation and a real-world system were performed.

Table 1. Summary of research works on robotic waste sorting.

Study	Robot	Gripper	Sorting Material	Simulation	Real System
[7]	Delta	Vacuum	MSW	No	Yes
[8]	Delta	Vacuum	MSW	Yes	Yes
[9]	Articulated	Soft Finger	MSW	No	Yes
[10]	Cartesian	Finger	CDW	No	Yes
[11]	Multi-robot Articulated	Finger	General Waste	No	Yes
[12]	Cartesian	Finger	General	No	Yes
[13]	Cartesian	Finger	CDW	No	Yes
[14]	Cartesian	Finger	CDW	No	Yes
[15]	SCARA	Not mentioned	General Waste	Yes	No
[16]	Articulated	Not mentioned	General Waste	Yes	No
[17]	Delta	Vacuum	MSW	No	Yes
[18]	Cartesian	Finger	CDW	No	Yes
[19]	Articulated	Finger	MSW	No	Yes
[20]	Articulated	Finger	CDW	No	Yes
[21]	Articulated	Vacuum	MSW	Yes	Yes

Diverse robotic configurations have been explored for waste sorting applications, each selected for their distinct motion capabilities and performance characteristics. Articulated or serial robots are the most popular choice due to their high dexterity and

flexibility [9,11,16,20]. The choice of a delta robot configuration is supported in [8] due to its high speeds, enabling higher efficiency and shorter cycle times. Cartesian robots are also a popular choice due to their rigidity, simple kinematics, and ability to carry large loads with high speeds [10]. For CDW sorting, articulated [20] and Cartesian robots [14,18] have been deployed in multiple cases.

When it comes to grippers, although there is no clear preference for a single type, suction and antipodal finger grippers are the most popular choices. There is, however, a significant correlation between application and gripper type. Indicatively, studies that develop sorting systems for municipal solid waste commonly utilize a vacuum gripper [7,8,17,21], while sorting solutions for CDW usually employ a finger gripper [10,13,18,20]. Unfortunately, comparisons between different robot and gripper types in simulation or real settings for the same application have not yet been examined in existing research.

In robotic sorting, especially in challenging environments such as CDW, it is common for the robot to perform its own motion planning after receiving target object information [11], resulting in limited control over the process. However, alternative motion planning schemes have also been explored, always relying on the kinematics of the used robot. In [10], a method for locating where the object should be grasped is calculated for a Cartesian robot under the assumption of a trapezoidal velocity profile. The motion planning for a particular delta robot in [7] relies on the solution of a differential equation with Newton's iterative method. Manipulation planning for sorting tasks closely resembles planning for pick-and-place, with the vast majority of studies favoring approaches where the end-effector aligns with a moving object, grasps it, moves above the suggested place position, and releases it. In [8,16], this manipulation approach was replaced with pick-and-throw (or pick-and-toss) alternatives. In [9], non-prehensile manipulation was compared with pick-and-place manipulation. These manipulation techniques resulted in faster, but less accurate cycles. Nevertheless, there is currently no universal planning method applicable to all types of robots in the literature of sorting systems.

Scheduling and decision-making regarding object prioritization have been explored in only a few studies, and there is no particular work examining CDW. One of the first works on waste sorting evaluated the most basic scheduling rules (FIFO, SPT) and provided improved versions of them [15]. In [12], a learning-based approach is used to predict the grasp and object that will give the highest purity of the output stream. In [21], the sequence of pick-and-place operations in a sorting system is optimized with a Genetic Algorithm (GA)-based approach.

Recent advances in learning-based robotic planning have introduced methods that leverage deep models for both grasping and scheduling. In particular, CNN-based architectures have shown promising results in vision-based grasp pose estimation, offering improved generalization and spatial reasoning [22,23]. These models learn to predict grasp configurations directly from RGB or RGB-D inputs and are increasingly used in industrial bin-picking or cluttered environments.

In parallel, Graph Neural Networks (GNNs), except for grasping and manipulation [24], have lately been proposed for difficult scheduling and combinatorial problems, particularly in complex, stochastic, and dynamic environments, with inevitable uncertainties. GNN-based policies have been shown to outperform heuristic and rule-based schedulers in domains such as job-shop scheduling, task assignment, and robotic fleet coordination [25,26].

Although simulation of a sorting waste system is able to provide important performance metrics for an application, it is generally not preferred in the literature [27]. On the other hand, modeling of the grasping process is difficult—simulating the rest of the process enables offline testing of any motion planning and decision-making parameters. In [8], the pick-and-throw manipulation approach was simulated in a V-REP environment to

assess its sorting rate. The sorting rate was also examined in [15] through simulations for different flow settings. Finally, dynamics simulations were used in [21] to determine the fitness function of the GA. For nearly all the studies, experiments with a real robotic component were performed, indicating the importance of an actual setup for sorting.

In summary, prior studies demonstrate a broad range of robotic configurations applied to waste sorting, with each work presenting a motion planning technique tailored to the chosen configuration. While scheduling and decision making is addressed in a few studies, this topic is always treated separately from motion planning. While learning-based approaches demonstrate strong task-specific and end-effector-specific performance, they often require large annotated datasets, extensive training, and can be less interpretable. Finally, simulation and real-world system setups are rarely developed at the same time; when they are, it is primarily to assess a specific planning choice. This work addresses these gaps by presenting RaapWaste, integrating both motion planning and scheduling. Furthermore, it supports simulation to allow fast development and testing, along with real-system integration for validation and verification. Building upon insights from prior research, this study aims to enhance performance for any robotic sorting application.

3. System Overview

This section provides a brief overview of robotic sorting systems, covering both hardware and software aspects to establish the context for the planning framework. In addition, the interaction among the different components is described in detailed and illustrated in Figure 1. Finally, the sorting system used for simulation and real experiments in later sections is presented.

Robotic sorting systems, like any cyber-physical system [28], integrate physical and computational components. Their key physical components are shown in Figure 2. Starting from the **conveyor**, it is responsible for moving materials through the sensor field of view and then the robot's workspace and it is usually controlled with a PLC and an encoder. For the real setup, a conveyor with a length of 10 m is utilized.

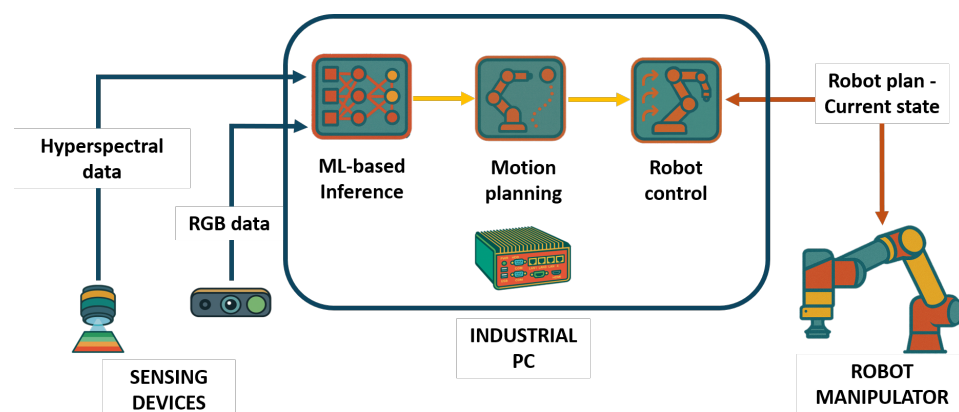


Figure 1. High-level flowchart of the information exchange among the system components.

The **optical sensors** provide positional data and assist with material classification. For the considered sorting system, an RGB camera (Basler acA1300-60gc, Basler AG, Ahrensburg, Germany) is utilized to provide positional information of the materials, while a hyperspectral camera (Specim FX17, Specim, Spectral Imaging Ltd, Oulu, Finland) assists in material classification.

An external PC manages the system's computational tasks, including sensory data processing, motion planning, and manipulator control. The PLC together with the PC constitute the **computational node** of the system.

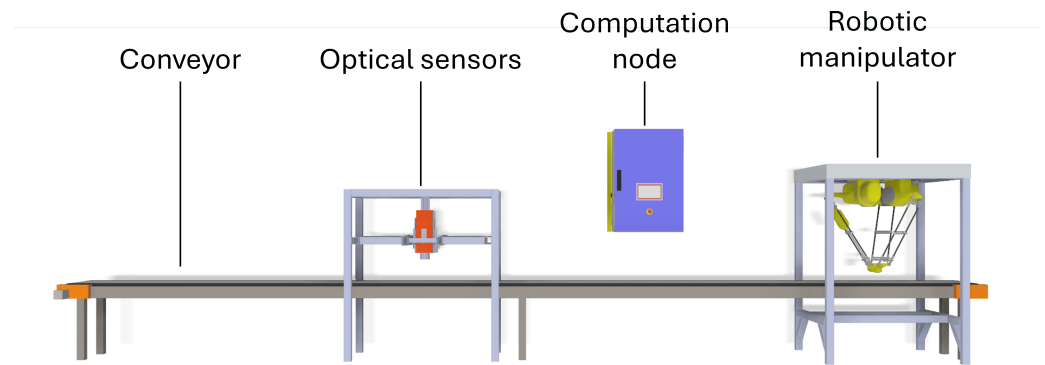


Figure 2. Main components of the sorting system: conveyor, optical sensors, computational node, and robotic manipulator.

Finally, the **robotic manipulator** is responsible for picking the material from the conveyor and placing them into predefined positions based on the processing of the sensory data already performed. The robot manipulator used in our setup is the UR5e by Universal Robots (Universal Robots A/S, Odense, Denmark). The gripper is a commercial two-finger gripper, made by OnRobot (OnRobot A/S, Odense, Denmark), whose fingers are covered with soft material to protect from wear.

Given the complexity of the system with multiple subsystems, a middleware solution was implemented for efficient data exchange. The Data Distribution Service (DDS) for ROS2 was selected to facilitate intra-process, inter-process, and inter-machine communication [29]. The entire software system, including grasp pose prediction, motion planning, and control, was designed for integration with the MoveIt2 stack [30], the leading robotic manipulation platform supported by ROS2.

The system's software components are implemented as ROS2 nodes, each focused on a specific task. These nodes are divided into two categories: (1) Those responsible for calculating grasp poses, and (2) those handling robot planning and control. The first category, which supports the planning process, is discussed here, while the second, representing the main contribution of this work, is covered in a separate section.

The system processes continuous data streams from the camera, feeding them into machine learning models within the Inference Pipeline for classification, localization, and segmentation tasks. The primary challenge in this pipeline is classification, where a multi-modal deep classifier is trained on a custom dataset until satisfactory accuracy (typically 95%) is achieved. Along with the classification, the pipeline provides positional information, including the object's contour and centroid, which are computed using a YOLOv8 model for real-time localization and segmentation [31].

Real samples from a CDW management facility were used to train the inference models and test the proposed planning solution. The samples were collected from a CDW stream that was suggested by the CDW management facility. This specific stream represents only a portion of the broader processing activities at the facility and it was chosen because it could not be further separated by conventional mechanical means. Therefore, it presents a highly relevant use case for robotic sorting, motivating the development and testing of our proposed framework.

The stream has undergone both crushing and shredding processes in order to homogenize and reduce the size of items, making them suitable for robot processing. It is mainly composed of stone, plastic, and wood materials, with compositions by weight shown in Table 2. However, this composition can vary significantly depending on the geographic location of the facility. Furthermore, there is batch-to-batch variability within the same facility due to fluctuations in the incoming waste materials.

Table 2. Properties of the CDW stream used for dataset generation.

Property	Value
Bulk density	150 kg/m ³
Material composition (in weight)	• Concrete and related stone materials: 60% • Plastic material: 21% • Wood: 15% • Cardboard: 3% • Metals: 1%
Grain size distribution	• 0–25 mm: 20% • 25–50 mm: 30% • 50–150 mm: 40% • >150 mm: 10%

For the dataset generation, 100 samples were collected from the stream, with the highest possible variability in materials and size. The sole criterion for including a sample from the CDW stream in the dataset is its size. Each sample size should be at least 25 mm in its largest dimension, while at least one dimension has to remain within the maximum allowable width of the finger gripper. Example images of the samples can be found in Figure 3.

**Figure 3.** Subset of the CDW dataset created.

To convert the positional data from pixel coordinates to camera frame coordinates, the system uses a geometric transformation based on the known dimensions of the camera's field of view on the conveyor belt [32]. The resulting geometric representation is then passed to the Grasp Pose Generator, which calculates the appropriate grasp poses based on the object's geometry and the selected gripper type.

This work focuses on two commonly used gripper types: vacuum and antipodal two-finger grippers. The sole use of a RGB camera restricts the grasps to a constant height component and constant orientation perpendicular to the conveyor belt plane. Within these constraints, simple heuristic-based methods are sufficient to generate viable grasp poses for a subset of objects.

While we acknowledge that more sophisticated grasp pose prediction algorithms, such as learning-based models, could significantly improve grasping robustness, their development and integration are beyond the scope of this work. Instead, our focus lies on evaluating and advancing the planning and scheduling components of the sorting pipeline, which are designed to accommodate improved grasping modules in future extensions.

For the vacuum gripper, the algorithm computes the grasp position by placing the center of the vacuum cup to the center of the mask that was generated by the localization and segmentation model. For the two-finger gripper, a heuristic algorithm is shown in Algorithm 1 determining the grasp position and orientation based on the object contour.

Algorithm 1 Heuristic Grasp Pose Generation algorithm for finger gripper.

Require: C (2D contour points of the object)

- 1: Compute the convex hull H of the 2D contour C .
 - 2: **for** each edge E in H **do**
 - 3: Compute the angle ϕ between E and X-axis.
 - 4: Rotate the H by $-\phi$.
 - 5: Compute the axis-aligned bounding box B of the rotated points.
 - 6: Record B if the area of B is the smallest so far.
 - 7: **end for**
 - 8: Extract the centroid M of the box.
 - 9: Determine the longer edge $\hat{E}$.
 - 10: Compute the angle $\hat{\phi}$ between $\hat{E}$ and the X-axis.
 - 11: Construct the grasp pose using:
 Position: M with fixed Z height,
 Orientation: Quaternion from $\hat{\phi}$ around Z-axis
-

4. Implementation

This section focuses on the development of RaapWaste for picking and placing the objects from the conveyor to the appropriate positions. Starting from the Scheduler, a feasible and optimal trajectory is derived given the grasp poses, while the Executor is responsible for the control based on the planned trajectory. Finally, the Sorting Controller implements the high-level control logic of the system in the form of Behavior Trees (BTs).

4.1. RaapWaste: Scheduler

Once the grasp pose of each object is available, the planning process of the robot motion follows. This complicated task is handled by the *scheduler*, which comes up with precise and feasible manipulation plans for picking and placing an object based on a specific scheduling rule or decision-making algorithm. Although each scheduler adheres to a different scheduling rule, the motions composing the manipulation plan and the planning stages required to compute them remain consistent. The following subsections detail the pick-and-place planning methodology and the scheduling rules that guide the overall process.

4.1.1. Pick-and-Place Pipelines

The manipulation plans for object picking and placing are generated through a structured pipeline designed to ensure reachability, collision-free paths, and temporal synchronization. Each stage incorporates specific computational processes to address the requirements of the task.

The picking process consists of the following stages:

- *Workspace and Reachability Analysis (Figure 4a)*: This combined stage checks if the grasp pose lies within the defined operating workspace and, second, if it is kinematically reachable and collision-free. Computationally, this involves verifying the grasp pose against geometric constraints of the robot and environment, and performing collision-checking using spatial models.
- *Approach and Tracking Motion*: The robot's trajectory to approach the grasp pose is computed. The planner calculates a trajectory path to reach the target object, including a linear translation along the conveyor (Figure 4c). This stage also ensures the approach path adheres to velocity and acceleration constraints for smooth motion.
- *Retreat Path Planning (Figure 4d)*: After grasp execution, a retreat motion is planned to lift the object from the conveyor or surface. This motion is typically vertical but may include additional lateral components to avoid obstructions. Trajectories are computed by incrementally sampling collision-free waypoints above the grasp pose.
- *Time Synchronization for Dynamic Grasping*: This stage computes a synchronization delay based on the relative velocities of the robot and the conveyor. By aligning the grasp pose with the predicted object position at a specific time, the system ensures accurate execution.

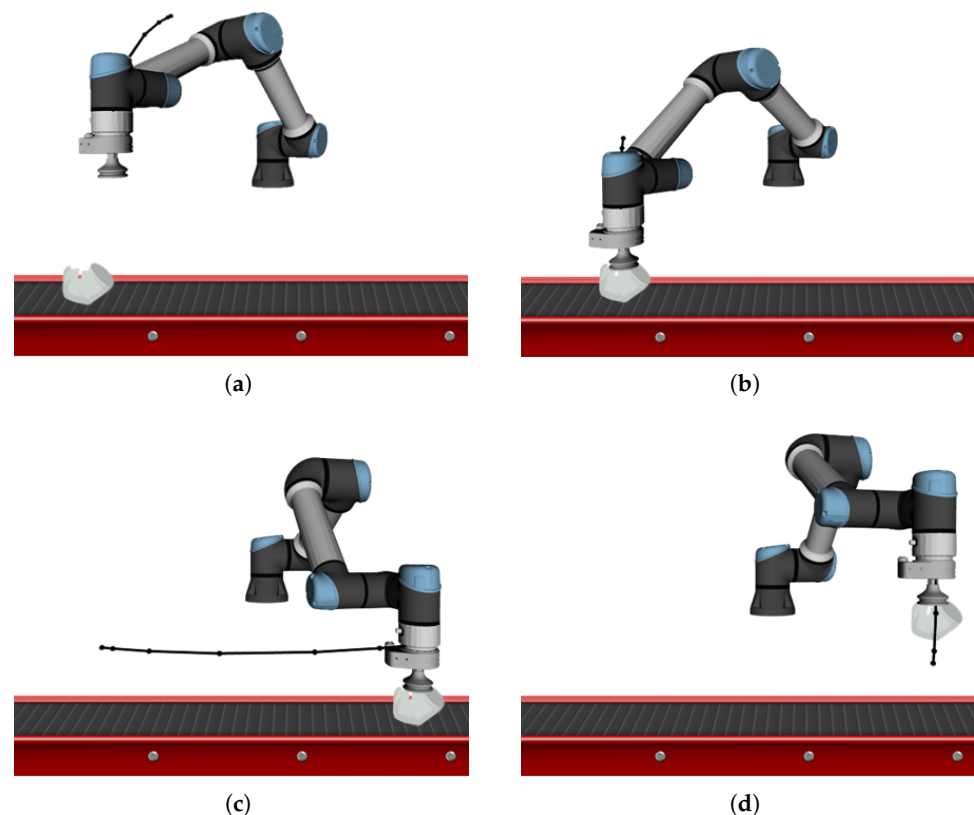


Figure 4. The manipulation plan for the picking process. (a) Reach above the object. (b) Approach. (c) Move linearly along the conveyor. (d) Lift the object from the conveyor.

The placing process is divided into the following stages:

- *Pose Validation and Reachability Analysis*: The placement pose is verified to ensure it is within the robot's workspace, kinematically reachable, and collision-free. This is performed using forward kinematics and collision-checking algorithms.
- *Approach and Retreat Planning*: Trajectories are computed to guide the end-effector toward the placement pose and retract it afterward. These motions may involve linear or curved paths, depending on task constraints, and the retreat phase may be skipped if unnecessary.
- *Pre-Placement Path Planning*: The robot computes a collision-free path to transition from its current configuration to a state aligned with the approach trajectory, ensuring a smooth and efficient motion into the placement phase.

For every plan computed during all these stages, collision-free feasibility is guaranteed, when a solution is found. However, the availability of a feasible solution also depends on the capabilities of the chosen path planners.

4.1.2. Scheduling Rules

The robot may be able to pick one or more objects at any given time. Thus, there needs to be a decision-making algorithm to prioritize the order in which objects are picked. These algorithms, commonly referred to in the literature as scheduling rules [15], aim to maximize process efficiency, i.e., the number of objects picked and placed per unit of time.

The most commonly used scheduling rules for PnP processes above a moving conveyor are First-In First-Out (FIFO) and Shortest Processing Time (SPT). These two heuristic rules are implemented as different Schedulers, though the general Scheduler framework easily accommodates the integration of other rules. Variations and examples of other rules given in the literature are Second-Shortest Processing Time (SSPT) [33], Largest Processing Time (LPT), Last-In First-Out (LIFO), Shortest Distance (SD), and Longest Distance (LD) [34]. As will be shown next, these scheduling rules can easily be implemented by performing minor modifications on the two algorithms of the FIFO and SPT rules. Nevertheless, inventing and comparing new and old scheduling rules is beyond the scope of this work.

The grasp poses created by the Grasp Pose Generator are stored in a queue Q as soon as they are received by the Scheduler. As an object moves above the conveyor, its grasp pose is translated together with the actual object. Therefore, there is no single grasp pose for each object, but multiple ones that can be reached at different points in time.

As part of this work, a method is proposed that projects the grasp poses into the future by translating them across the conveyor's movement axis in discrete increments (see Figure 5). Each one is then evaluated by the picking pipeline, and if it qualifies as a solution, the manipulation plan is considered a candidate. Finally, multiple grasp poses are evaluated at the same time with a multi-threading design of all the Schedulers, thus accelerating solution of the problem.

The implementation of the planning pipeline using the FIFO scheduling algorithm is shown in pseudocode in Algorithm 2. Starting with queue Q , each grasp pose is evaluated sequentially in a First-In First-Out (FIFO) manner, meaning that grasp poses corresponding to objects that have passed through the sensor unit earlier are processed first. For each grasp, the translated grasps are placed in a new queue Q_t , where poses inserted later have larger translations. For each translated grasp pose p_t in Q_t , the pipeline stages are evaluated sequentially. If a valid manipulation plan M is found, then both loops stop without evaluating further poses.

Although it will not be presented in this work, the LIFO scheduling rule can be written as a variation of FIFO when the queue Q is replaced with a stack S . Then, the lat-

est poses will be processed first and as soon as a valid manipulation plan M is found, the algorithm stops.

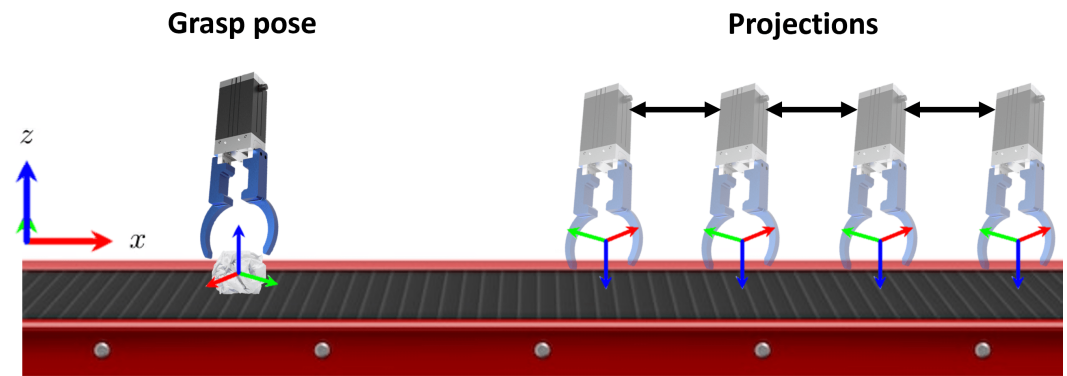


Figure 5. Original grasp pose and its projections into the future.

Algorithm 2 Derivation of pick manipulation plan using FIFO scheduling rule.

Require: Q

```

1: for  $p$  in  $Q$  do
2:    $Q_t \leftarrow \text{GenerateTranslatedPoses}(p)$ 
3:   for  $p_t$  in  $Q_t$  do
4:      $M \leftarrow \text{CreateManipulationPlan}(p_t)$ 
5:     if  $M$  is valid then
6:       Break/Exit both loops
7:     end if
8:   end for
9: end for

```

In Algorithm 3, the SPT implementation is described. In this case, all the grasping poses need to be processed. If a valid manipulation plan M is found for a pose p_t , then it is compared with the shortest manipulation plan until that point. If its time duration is smaller, then it replaces the previous shortest manipulation plan as the optimal one.

The SSPT and LPT scheduling rules have the same implementation as the SPT, but now instead of comparing and keeping the currently shortest manipulation plan, the plan with the second shortest and the largest duration will be kept.

Algorithm 3 Derivation of pick manipulation plan using SPT scheduling rule.

Require: Q

```

1:  $\text{minTimeDuration} \leftarrow +\infty$ 
2:  $\text{shortestManipulationPlan} \leftarrow \text{NULL}$ 
3: for  $p$  in  $Q$  do
4:    $Q_t \leftarrow \text{GenerateTranslatedPoses}(p)$ 
5:   for  $p_t$  in  $Q_t$  do
6:      $M \leftarrow \text{CreateManipulationPlan}(p_t)$ 
7:     if  $M$  is valid then
8:        $\text{duration} \leftarrow \text{CalculateTimeDuration}(M)$ 
9:       if  $\text{duration} < \text{minTimeDuration}$  then
10:         $\text{minTimeDuration} \leftarrow \text{duration}$ 
11:         $\text{shortestManipulationPlan} \leftarrow M$ 
12:      end if
13:    end if
14:  end for
15: end for

```

4.1.3. Computational Complexity

In the previous section, we presented the implementations of two classical scheduling heuristics for our work: FIFO and SPT. However, each offers distinct trade-offs in computational complexity and scheduling behavior. To analyze their complexity, consider the following parameters:

- $n = |Q|$: Number of grasp poses in the main queue.
- $k = |Q_i|$: Number of translated poses per grasp (assumed constant per grasp).
- c : Cost of generating a single manipulation plan (*CreateManipulationPlan*). The cost is not constant for every iteration, because the process may fail at an early stage of the planning pipeline (see Section 4.1.1).
- d : Cost of *CalculateTimeDuration* for a valid plan, which can be considered negligible for our cases, because we have already calculated the entire manipulation plan.

In Table 3, the computational complexity of the two Schedulers is presented.

Table 3. Computational complexity for the two schedulers FIFO and SPT.

Scheduler	Worst-Case Complexity	Best-Case Complexity	Insertion time
FIFO	$O(n \times k \times c)$	$O(c)$	$O(1)$
SPT	$O(n \times k \times c)$	$O(n \times k \times c)$	$O(1)$

The computational complexity of the FIFO-based planning algorithm depends on the number of grasp poses n , the number of translated versions k per grasp pose, and the cost c of generating a manipulation plan. In the worst-case scenario—where no valid grasp is found—the algorithm evaluates all $n \times k$ candidates, resulting in a complexity of $O(n \times k \times c)$. In the best case, a valid manipulation plan is found immediately, leading to a complexity of $O(c)$. The early termination condition improves efficiency in practice.

The SPT-based planning algorithm evaluates all grasp poses and selects the one associated with the shortest manipulation time. For n grasp candidates, each with k translated poses, the worst-case computational complexity is $O(n \times k \times c)$, where c is the cost of generating a manipulation plan. Unlike FIFO, which may terminate early, the SPT approach requires full evaluation of all candidates to determine the optimal plan, resulting in higher but more predictable computational cost.

The computational overhead required to evaluate all the grasp poses is reduced with temporal filtering, which is periodically implemented to prune old grasps of the queue Q . This drastically reduces the size of Q of every iteration. Furthermore, concurrency is leveraged by spawning multiple processing threads at a time, each responsible for evaluating a translated pose p_i . This way, the computation of a valid manipulation plan is further accelerated.

4.2. RaapWaste: Executor

As soon as a manipulation plan is recovered from the Scheduler, an Executor is responsible for controlling the motion of the robot at higher-level and to ensure the plan is followed.

The Executor forwards the trajectories to the low-level controller of the robot, without further processing, and monitors the completion of the individual motions. Any large disturbance in the monitored operation causes immediate stop of the robot motion.

Except for the control of the robot, the Executor handles the control of the end-effectors, sending the necessary commands to a common interface developed for all supported grippers. Such commands can be a setpoint distance for the finger gripper or a target vacuum pressure.

The framework of Executors is designed also to support the hybrid deliberative/reactive (*Plan, Sense-Act*) paradigm [35], where a local planner acts alongside a global one and adjusts the robot motion in real-time based on a sensory input. In future work, a reactive Executor will be implemented to grasp objects with greater accuracy based on a vision sensor mounted on the robot.

5. Experimental Results

In this section, the RaapWaste framework is evaluated through a series of experiments conducted in simulation and a real-world sorting system. Starting with simulations of two robots of different types, the generality of the approach is shown. Then, a CDW sorting case-study is evaluated in a real-world setup, showing (i) the transferability of simulation to real-world system, and (ii) the performance of the planning approach in a realistic use.

In the following experimental evaluations, we assess the performance of the Raap-Waste framework using three primary metrics: Sorting Throughput (ST), Sorting Ratio (SR), and Grasping Success (GS).

- The **Sorting Throughput (ST)** is defined as the number of objects picked and placed per minute and reflects the system's cycle efficiency under varying conditions.
- The **Sorting Ratio (SR)** refers to the percentage of detected objects for which a complete manipulation plan (including approach, pick, and place) is successfully executed. It is important to note that successful plan execution does not necessarily imply a successful grasp, as physical failures, such as missed or unstable grasps, may still occur even when a valid motion plan is followed.
- The **Grasping Success (GS)** is defined as the percentage of objects that were successfully grasped relative to the total number of objects that were approached by the robot (i.e., for which a motion plan was executed and a grasp attempt was made). GS is only evaluated for the real-world experiments.

5.1. Simulated Experiments

The simulated scenario is a sorting application where objects of two waste types are separated in two different bins. The objects are generated randomly, with their positions and waste types following different normal and uniform distributions, respectively. The spawning times of each object are generated through a Poisson distribution.

Two different robots were chosen to be simulated: (i) a UR5e collaborative robot by Universal Robots with 6-DOF, and (ii) a delta FANUC M-3iA/6S robot (FANUC Corporation, Oshino-mura, Yamanashi, Japan) with 4-DOF (see Figure 6). The former offers dexterity, while the latter is able to move at high speeds.

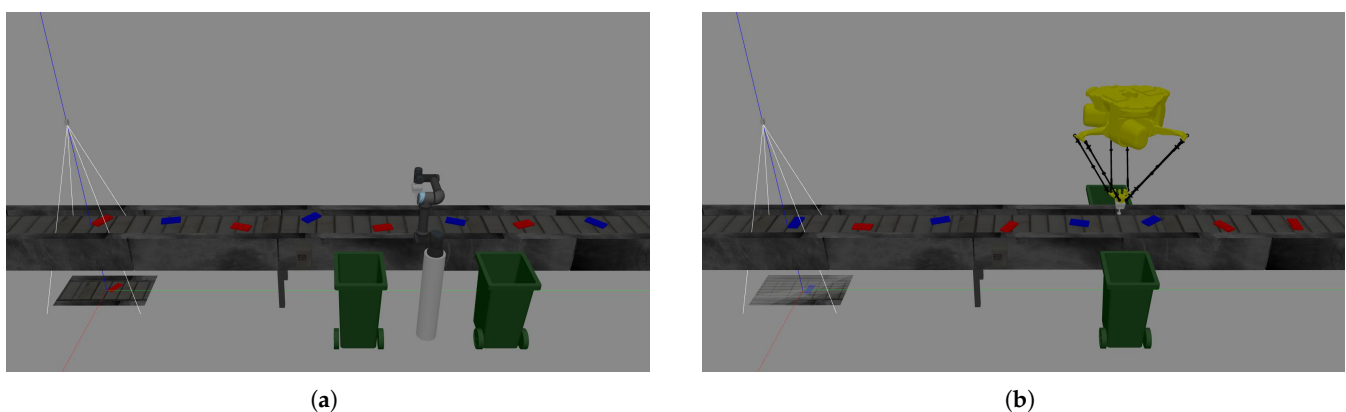


Figure 6. Simulation of a waste sorting system with a UR5e articulated robot (a) and a FANUC M-3iA/6S delta robot (b).

The kinematics of both robots can be analyzed analytically and solved rapidly [36]. However, certain kinematic parameters are not readily available, because robot manufacturers withhold this information. Therefore, realistic estimates are provided in place of exact values, but exact cycle times are not expected, especially for the delta robot.

Due to the seamless integration with ROS and a variety of industrial robotic hardware, Gazebo was chosen as the simulation environment for this work. Given the complexity of modeling the gripping and suction dynamics of this application, picking is simplified by attaching the target object to the gripper once certain geometric constraints are met. In most cases, these constraints require that the gripper and the object are in contact and their centers of mass are close. For these experiments, a suction gripper is simulated.

To assess the efficiency of the proposed planning solution, simulation experiments lasting 600 s were performed for both robots. In each trial, the objects are placed on the conveyor with a different mean rate, which is referred to as the Placement Rate (PR) from now on and is measured in objects per minute. The conveyor velocity is fixed at 0.35 m/s. Figures 7 and 8 show these metrics for each robot for different mean spawning intervals.

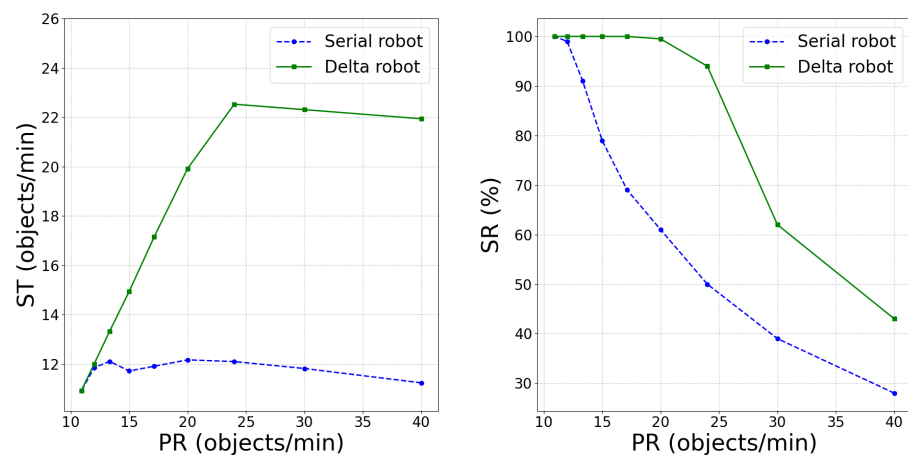


Figure 7. Sorting metrics of serial and delta robots in simulation using FIFO scheduling rule.

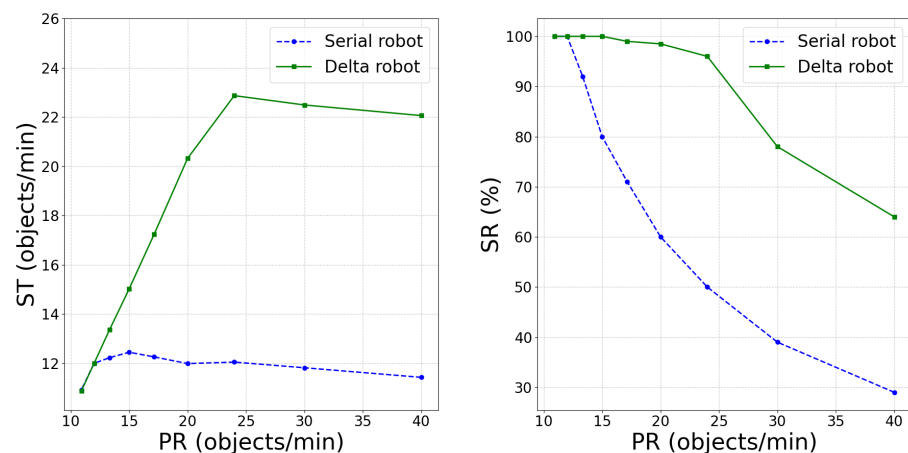


Figure 8. Sorting metrics of serial and delta robots in simulation using SPT scheduling rule.

5.2. Real-Robot Experiments: CDW Sorting

The robot manipulator in this setup is UR5e by Universal Robots, which is the same as the one used in previous simulations. The gripper is a commercial two-finger gripper, made by OnRobot, whose fingers are covered with a soft material to protect them from wear. The UR5e robot was placed sideways on the conveyor and the conveyor velocity was

set for all experiments at 0.35 m/s. The utilized manipulator, finger gripper, and conveyor are shown in Figure 9.



Figure 9. UR5e Robot, finger gripper, and conveyor performing CDW sorting experiments.

We evaluated the planning and scheduling approach through a series of real-world experiments using the CDW samples shown in Figure 3. Like the simulations in Section 5.1, these trials varied the Placement Rate (PR) and scheduler configuration. In each case, we measured the Sorting Throughput (ST) and Sorting Ratio (SR) to assess performance.

More detailed metrics are shown in Table 4. The four rightmost columns show the number of objects detected by the Inference Pipeline, the number of objects grasped and sorted, the number of objects correctly approached by the robot but missed, and finally, the SR. The ST and SR metrics for each trial are plotted in Figure 10.

Since this work intends to assess the planning efficiency and not the gripping or detecting efficiency, the SR is the ratio of the total number of approached objects (although some may be missed) to the total number of detected objects.

Table 4. Sorting metrics for the two schedulers and different mean placing intervals.

Scheduler	PR (Objects/min)	Total Objects	Detected Objects	Sorted Objects	Approached But Missed Objects	Sorting Ratio (%)	Grasping Success (%)
FIFO	20	100	100	50	6	56.0	89.3
FIFO	15	100	97	67	7	76.3	90.5
FIFO	12	100	99	87	7	94.9	92.6
SPT	20	100	100	52	6	58.0	89.7
SPT	15	100	100	73	7	80.0	91.3
SPT	12	100	98	90	7	99.0	92.8

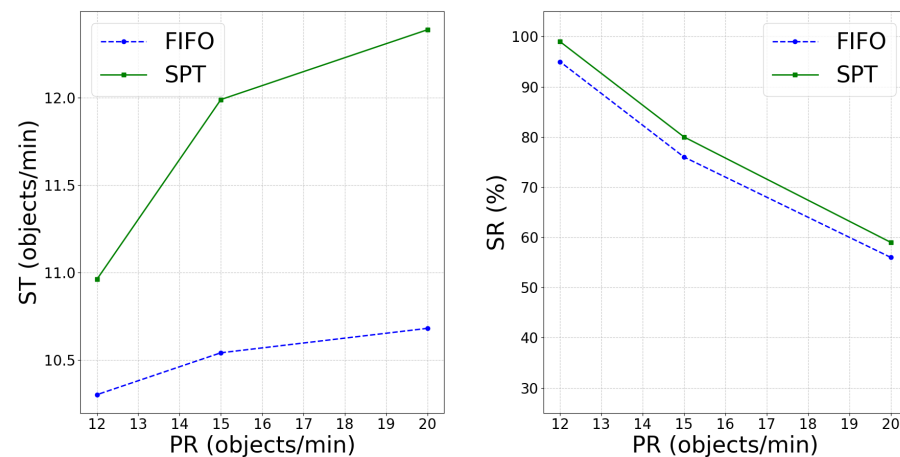


Figure 10. Sorting metrics for the CDW stream.

6. Discussion

6.1. Evaluation of Performance Metrics

The observed trade-off between the Sorting Ratio (SR) and Sorting Throughput (ST) presents a fundamental challenge in real-world robotic sorting. Higher Placement Rates (PR) increase object availability, reducing idle time and improving throughput. However, this also decreases the available planning time per object, which may lead to missed grasps and lower SR.

In practice, the relative importance of SR and ST depends on the operational goals of the facility. For instance, when sorting high-value materials or targeting purity requirements for reintegration into production, maximizing SR may be critical—even at the cost of reduced throughput. Conversely, in throughput-driven environments where volume recovery is prioritized, a lower SR may be acceptable if more total material is processed per minute.

A comparison of performance metrics between simulation and real-world setup reveals no significant discrepancies for the serial robot, particularly with regard to the sorting ratio. This confirms that the performance observed in simulation reliably reflects real-world behavior.

This conclusion supports the use of RaapWaste in simulation not only for testing, but also for reliably predicting system behavior under realistic conditions. This capability provides stakeholders in both research and industry with the flexibility to conduct comprehensive, end-to-end evaluations of robots, planners, and grippers before committing to a real-world implementation.

6.2. Comparison of Scheduling Rules for the CDW Sorter

The performance of schedulers SPT and FIFO in simulation is shown in Figures 7 and 8, while Table 4 provides insight for the real-world setup.

The metrics for both schedulers suggest a slight improvement in performance when the SPT scheduler is preferred. The robot is able to grasp more objects in the same time frame when SPT scheduling rule is used for every PR tested, while the ratio of sorted objects is always higher for SPT. This observation is validated by both simulated and real-world experiments.

For the real-world setup, SPT also has a higher SR and ST, since it is able to pick more objects in the unit of time, while the approached but missed objects are equally frequent for both schedulers, showing that the scheduler choice does not affect the validity of the motion plan derived.

Their difference arises because the SPT scheduler typically comes up with shorter travel distances for the robot. On the other hand, planning with FIFO is faster than planning with SPT, since it does not require to process all the objects before making a decision on the object to be picked.

Although only these two schedulers were tested, they can be replaced by any other heuristic or learning-based scheduler without further changes due to the modular nature of the planning framework. The system integrator can exploit any available insight and develop the most efficient scheduler for the task of interest.

6.3. Comparison of Robots for the CDW Sorter

The performance metrics for both robots used in simulation, Universal Robots UR5e and FANUC M-3iA/6S delta, are shown in Figures 7 and 8. The delta robot is able to perform more cycles than the articulated robot, as expected, resulting in a higher ST. Since the delta robot sorts more objects, fewer objects are missed and a higher SR is reached in every case.

The delta robot achieves its maximum ST when 24 objects are given as input on average per minute, while the ST of the serial robot is maximized for 15 and 20 objects per minute for the FIFO and SPT schedulers, respectively. When the placement rate is reduced from its maximum, fewer objects are available for the robot to pick, resulting in a lower ST. Conversely, increasing the placement rate above its maximum slightly reduces the ST, as the planning time per cycle increases due to the larger number of objects to process.

Although real-world experiments are not performed for the delta robot, a delta robot similar to FANUC M-3iA/6S with four DOF will result in superior performance than the Universal Robots UR5e, given that the target grasp poses are attainable by the underactuated robot. Due to the design choices made while developing the RaapWaste framework, integration and real-world deployment of the delta robot should have a well-defined overhead, leading to predictable performance metrics.

6.4. Comparison with Prior Work

In this section, a comparison with similar state-of-the-art sorting systems is performed to assess the contribution of the RaapWaste framework. For consistency, we include only works that report robot-related performance metrics. Many studies in the literature focus on perception and provide exclusively inference-related metrics, as in [11]. These works are excluded from this analysis, as the focus of our work lies in the planning and control aspects of robotic sorting. Furthermore, studies authored by the same groups and employing the same setups for similar waste materials are grouped together to avoid redundancy. For example, refs. [10,13,14] describe the same setup and provide very similar metrics.

In Table 5, all the metrics from the aforementioned works are reported, along with the sorting materials and robot and gripper types. A first observation is that only some of the previously cited works provide relevant metrics for their proposed sorting systems. Moreover, metrics such as the Sorting Throughput (ST) and Sorting Ratio (SR), which are particularly relevant for evaluating the planning component, are sparsely reported. This further underscores the need for more structured benchmarking practices in the development of robotic sorting systems.

Unlike the other studies, which only evaluate a single robotic configuration, RaapWaste performance can be demonstrated across both serial and delta robots, showcasing its robot-agnostic design. This is particularly important for industrial deployment, where hardware heterogeneity is common.

RaapWaste achieves Sorting Throughput of 12.5 and 22.5 objects/min for the serial and delta robot, respectively. However, these metrics are highly dependent on the robot

type and robot model. Thus, the results of our serial robot experiments are compared with the only other serial robot study identified in the literature, i.e., [9], which reports experiments using two manipulation methods: pick-and-release (left value in the table) and push-and-drop (right value).

It is observed that our robot performs faster and achieves higher cycle success when compared with the pick-and-release method. Moreover, when compared to the push-and-drop approach, RaapWaste shows a significantly higher Grasping Success, indicating that the push-based strategy may be unsuitable for reliable use in CDW sorting scenarios.

When the performance of our simulated delta robot is compared with the reported results of studies that have integrated faster hardware, i.e., delta and Cartesian robots, we observe similar outcomes in terms of the Sorting Ratio. However, the Sorting Throughput (ST) is lower than the corresponding values reported in those cases. This indicates the influence of hardware limitations on cycle rates, such as the actuator speed and manufacturer-optimized motion profiles.

It is important to emphasize, however, that RaapWaste is designed as a planning and scheduling framework, not a robot-specific performance benchmark. Despite using a delta robot with slower dynamics, our framework was able to achieve high sorting accuracy, validating its robustness and adaptability.

Table 5. Comparison of RaapWaste with representative state-of-the-art robotic sorting systems from the literature.

Study	Robot	Gripper	Sorting Material	ST (Objects/min)	SR (%)	GS (%)
[7]	Delta	Vacuum	MSW	-	-	82–90
[8]	Delta	Vacuum	MSW	High	-	87–94
[9]	Serial	Soft Finger	MSW	11.5/18	-	70/57
[10]	Cartesian	Finger	CDW	35	92–96	96
[15]	SCARA	-	General Waste	-	97 (max)	-
RaapWaste	Serial/Delta	Finger/Vacuum	CDW	12.5/22.5	99 (max)	88–92

Cells with—denote missing information.

6.5. Integration Steps of a New Robot

Since the main focus of this work lies on the robot agnostic and adaptable nature of the framework, the overhead of integrating a new robot should be evaluated. Below we enumerate the necessary steps and changes to integrate a new robot, given that the rest of the setup, i.e., sensors, conveyor, and computation node remains the same.

1. **Perform a workspace analysis of the robot.** Given the kinematics of the robot, its useful workspace for each use-case is defined. The values of the kinematics parameters are in most cases available by examining the CAD files of the robot.
2. **Derive an inverse kinematics solution.** The inverse kinematics are necessary to transform robot poses to joint configurations. The vast majority of non-redundant industrial robots, like Universal Robots UR5e and FANUC M-3iA/6S, also have an analytic and efficient solution.
3. **Use an ROS control driver for continuous position or velocity control of the robot.** Most robot manufacturers like Universal Robots and COMAU support many of their models, while the ROS community provides working drivers for other vendors like ABB and FANUC [37].

4. **Perform hand-eye calibration.** Due to the new positioning of the robot, hand-eye calibration will be required to calculate the transformation from the sensor to the robot world frame.

Since the gripper is controlled by the computation node and not the robot controller, no additional steps are required when the robot changes, but the gripper remains the same

6.6. Extension to Multi-Robot Setups

Our experiments were limited to single-arm systems, and scaling to multi-arm setups has not been tested or evaluated. Multi-robot systems are often deemed necessary for higher throughput in waste sorting, but require additional planning and coordination among the different agents. Since RaapWaste aspires to remain agnostic to the robot types and number, while favoring robustness, scalability, and flexibility, a decentralized planning method is considered a better solution for a multi-robot configuration.

Decentralized scheduling with rules like SPT and FIFO can (i) be implemented with minimal computational overhead, (ii) distributed across separate computational nodes, and (iii) easily accommodate additional robots without requiring significant redesign of the system. Coordination can be added through lightweight mechanisms, such as grasp reservations or object claiming, if needed.

The assignment of the scheduling rule to each robot can be performed with a meta-heuristic algorithm like GRASP as performed in [34]. RaapWaste can assist this calculation of an optimal set of rules in simulation through the easy integration of a new robot.

6.7. Industrial Deployment and Real-World Limitations

Several practical integration challenges related to industrial deployment have been identified while designing and developing the RaapWaste framework. In Table 6, we describe the measures taken to address some of these challenges and outline the remaining limitations, along with potential strategies for tackling them in future implementations.

Finally, although simulation played a key role in our development and validation, especially with the delta robot, it is acknowledged that discrepancies may exist between simulated and real-world cycle times, due to missing low-level dynamic parameters by robot manufacturers.

Table 6. Integration challenges, RaapWaste responses, and existing limitations.

Challenge	RaapWaste	Limitation
Real-Time Constraints Robotic sorting systems must meet strict cycle-time requirements, since delays in planning or grasping can cause bottlenecks.	Implements early pruning and temporal filtering to reduce grasp evaluation load. Parallel processing is used for pose evaluations, while deterministic scheduling rules (e.g., FIFO, SPT) ensure bounded execution time.	
Uncertainty in Object Detection and Pose Estimation Unexpected disturbances such as cluttered scenes, occlusions, or sensor noise can reduce reliability.	The robot follows pre-planned trajectories without mid-execution corrections. Deterministic scheduling ensures repeatable behavior.	The lack of reactive feedback during grasp execution could limit reliability in an industrial setting

Table 6. Cont.

Challenge	RaapWaste	Limitation
Object Diversity and Deformability Items can be irregular, deformable, or overlapping.	Uses a heuristic grasp generator based on 2D object contours. Simulation is based on simplified grasping dynamics	Performance may degrade in cluttered, occluded, or noisy scenes, underscoring the need for more robust grasp pose generation and realistic grasping dynamics in future work.
Robot Workspace and Reachability Constraints Real robotic cells have physical constraints and obstructions.	Integrates kinematics of different robots and uses MoveIt2 for collision checking during pose evaluation.	
Unavailable Kinematic Parameters Industrial robots may not provide accurate kinematic models, especially velocities and accelerations	Supports integration via manually defined kinematics or reverse-engineered approximations. It uses MoveIt2 for validation and pose planning.	Imprecise models may reduce planning accuracy and require vendor support or calibration for reliable deployment.
Conveyor and Flow Integration Real setups may exhibit variable conveyor speeds and unsteady object arrival rates.	Schedulers estimate future object positions and integrate conveyor velocity to predict arrival times within graspable zones.	
Temporal Variability of Input Materials Material variability (shape, size, type) can affect sorting performance and shift priorities.	Uses deterministic rules to provide predictable decisions, but performance may degrade under high variability and changing conditions.	Adaptive decision-making and planning would better handle shifting priorities, evolving sorting criteria and unexpected disturbances such as cluttered scenes, occlusions or sensor noise.
Maintainability and Data Logging Long-term maintenance, debugging, and system traceability are required for industrial adoption.	Employs modular ROS2 plugins and uses ROS2 logging system to record all planning and execution steps, supporting debugging, monitoring, and continuous improvement.	
Safety and Certification Safety compliance with industrial standards is essential.	Designed to complement built-in safety features of industrial robots (e.g., force limits, safe zones, E-stop circuits) without interfering with them.	

7. Conclusions and Future Work

The increased complexity of CDW sorting operations underscores the critical need for innovative robotic planning architectures to ensure efficiency, adaptability, and reliability. This work addressed these challenges by introducing a novel planning solution tailored for robotic CDW sorting applications. The contributions of this research span multiple dimensions, offering a comprehensive approach to overcoming key limitations in existing systems.

Firstly, the proposed planning framework is robot-agnostic, enabling its deployment across various robotic configurations and applications. This flexibility is achieved by decoupling the motion planning and scheduling processes from specific robot types and their kinematics, broadening the solution's applicability.

Another key innovation in this work is the modular design, which accommodates integration of various scheduling rules, including FIFO and SPT, while enabling efficient evaluation of grasp poses through a projection-based method. Additionally, the Scheduler-

Executor paradigm separates the planning process from the robot's motion control, allowing each to operate independently.

The integration of ROS2 and MoveIt2 frameworks accommodates the incorporation of novel algorithms for planning, kinematics, and control, ensuring the system's scalability and future extensibility. The modularity and adaptability of BTs allow for efficient error handling and recovery, ensuring robust operation in dynamic and unpredictable environments. Furthermore, the inclusion of planning pipelines for both picking and placing tasks ensures the generation of collision-free and optimal trajectories, thereby further enhancing the reliability of the system.

Experiments were conducted to validate the proposed planning solution. Simulations evaluated the system's performance using both a Universal Robots UR5e collaborative robot and a FANUC M-3iA/6S delta robot. Real-world experiments focused on CDW sorting using a Universal Robots UR5e robot and a two-finger gripper. ST and SR were analyzed under different placement rates and schedulers, confirming the framework's adaptability across configurations, while real-world metrics remained consistent with those observed in the simulation.

In conclusion, this work presents a versatile and robust planning solution for robotic waste sorting systems. By integrating innovative approaches in motion planning and scheduling, the framework not only advances the state of the art, but also lays a solid foundation for future research and development. Future work will focus on enhancing the framework's capabilities by incorporating (i) reactive control, (ii) additional scheduling rules, and (iii) 3D perception into the grasp pose generation, (iv) exploring learning-based planning algorithms, and (v) extending the framework to multi-robot configurations.

Author Contributions: Conceptualization, K.K., F.K.K., and M.K.; methodology, K.K., F.K.K., and M.K.; software, K.K.; validation, K.K.; formal analysis, K.K.; investigation, K.K.; resources, F.K.K., G.T., and A.A.; data curation, K.K.; writing—original draft preparation, K.K.; writing—review and editing, F.K.K., M.K., and P.F.; visualization, K.K. and M.K.; supervision, M.K. and P.F.; project administration, F.K.K. and G.T.; funding acquisition, G.T. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the European Union's Horizon Europe research and innovation program under grant agreement No. 1010916 (project REDOL).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CDW	Construction and Demolition Waste
ST	Sorting Throughput
SR	Sorting Rate
PR	Placement Rate
BT	Behavior Trees
SPT	Shortest Processing Time
FIFO	First-In First-Out
SSPT	Second-Shortest Processing Time

LPT	Largest Processing Time
LIFO	Last-In First-Out
SD	Shortest Distance
LD	Longest Distance
DDS	Data Distribution Service

References

1. Soto-Paz, J.; Arroyo, O.; Torres-Guevara, L.E.; Parra-Orobio, B.A.; Casallas-Ojeda, M. The circular economy in the construction and demolition waste management: A comparative analysis in emerging and developed countries. *J. Build. Eng.* **2023**, *78*, 107724. [\[CrossRef\]](#)
2. Zhang, Z.; Xu, C.; Cheng, G.; Lau, E.V. Towards carbon neutrality: A comprehensive study on the utilization and resource recovery of coal-based solid wastes. *Int. J. Coal Sci. Technol.* **2025**, *12*, 34. [\[CrossRef\]](#)
3. Cao, S.; Che, C.; Zhang, Y.; Shan, C.; Liu, Y.; Zhao, C.; Du, S. Mechanical properties and damage evolution characteristics of waste tire steel fiber-modified cemented paste backfill. *Int. J. Min. Sci. Technol.* **2024**, *34*, 909–924. [\[CrossRef\]](#)
4. Liang, S.; Gu, Y. A deep convolutional neural network to simultaneously localize and recognize waste types in images. *Waste Manag.* **2021**, *126*, 247–257. [\[CrossRef\]](#)
5. Kiyokawa, T.; Takamatsu, J.; Koyanaka, S. Challenges for Future Robotic Sorters of Mixed Industrial Waste: A Survey. *IEEE Trans. Autom. Sci. Eng.* **2022**, *21*, 1023–1040. [\[CrossRef\]](#)
6. Lu, W.; Chen, J. Computer vision for solid waste sorting: A critical review of academic research. *Waste Manag.* **2022**, *142*, 29–43. [\[CrossRef\]](#)
7. Koskinopoulou, M.; Raptopoulos, F.; Papadopoulos, G.; Mavrikakis, N.; Maniadakis, M. Robotic Waste Sorting Technology: Toward a Vision-Based Categorization System for the Industrial Robotic Separation of Recyclable Waste. *IEEE Robot. Autom. Mag.* **2021**, *28*, 50–60. [\[CrossRef\]](#)
8. Raptopoulos, F.; Koskinopoulou, M.; Maniadakis, M. Robotic Pick-and-Toss Facilitates Urban Waste Sorting. In Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), Hong Kong, China, 20–21 August 2020; pp. 1149–1154. [\[CrossRef\]](#)
9. Kiyokawa, T.; Katayama, H.; Tatsuta, Y.; Takamatsu, J.; Ogasawara, T. Robotic Waste Sorter With Agile Manipulation and Quickly Trainable Detector. *IEEE Access* **2021**, *9*, 124616–124631. [\[CrossRef\]](#)
10. Ku, Y.D.; Yang, J.H.; Fang, H.Y.; Xiao, W.; Zhuang, J.T. Optimization of Grasping Efficiency of a Robot Used for Sorting Construction and Demolition. *Int. J. Autom. Comput.* **2020**, *17*, 1751–8520. [\[CrossRef\]](#)
11. Zhihong, C.; Hebin, Z.; Yanbo, W.; Binyan, L.; Yu, L. A vision-based robotic grasping system using deep learning for garbage sorting. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 11223–11226. [\[CrossRef\]](#)
12. Kujala, J.V.; Lukka, T.J.; Holopainen, H. Classifying and sorting cluttered piles of unknown objects with robots: A learning approach. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 971–978. [\[CrossRef\]](#)
13. Xiao, W.; Yang, J.; Fang, H.; Zhuang, J.; Ku, Y.; Zhang, X. Development of an automatic sorting robot for construction and demolition waste. *Clean Technol. Environ. Policy* **2020**, *22*, 1829–1841. [\[CrossRef\]](#)
14. Ku, Y.; Yang, J.; Fang, H.; Xiao, W.; Zhuang, J. Deep learning of grasping detection for a robot used in sorting construction and demolition waste. *J. Mater. Cycles Waste Manag.* **2021**, *23*, 84–95. [\[CrossRef\]](#)
15. Mattone, R.; Divona, M.; Wolf, A. Sorting of items on a moving conveyor belt. Part 2: Performance evaluation and optimization of pick-and-place operations. *Robot. Comput.-Integr. Manuf.* **2000**, *16*, 81–90. [\[CrossRef\]](#)
16. Chen, H.; Zhang, B.; Fuhlbrigge, T. Robot Throwing Trajectory Planning for Solid Waste Handling. In Proceedings of the 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Suzhou, China, 29 July–2 August 2019; pp. 1372–1375. [\[CrossRef\]](#)
17. Lin, Y.H.; Mao, W.L.; Fathurrahman, H.I.K. Development of intelligent Municipal Solid waste Sorter for recyclables. *Waste Manag.* **2024**, *174*, 597–604. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Li, Z.; Deng, Q.; Liu, P.; Bai, J.; Gong, Y.; Yang, Q.; Ning, J. An intelligent identification and classification system of decoration waste based on deep learning model. *Waste Manag.* **2024**, *174*, 462–475. [\[CrossRef\]](#)
19. Zhang, Z.; Wang, H.; Song, H.; Zhang, S.; Zhang, J. Industrial Robot Sorting System for Municipal Solid Waste. In *Intelligent Robotics and Applications*; Yu, H.; Liu, J.; Liu, L.; Ju, Z.; Liu, Y.; Zhou, D., Eds.; Springer: Cham, Switzerland, 2019; pp. 342–353. [\[CrossRef\]](#)
20. Lukka, T.J.; Tossavainen, T.; Kujala, J.V.; Raiko, T. *ZenRobotics Recycler—Robotic Sorting using Machine Learning*; ZenRobotics Ltd.: Helsinki, Finland, 2014.

21. Gundupalli, S.P.; Shukla, R.; Gupta, R.; Hait, S.; Thakur, A. Optimal Sequence Planning for Robotic Sorting of Recyclables From Source-Segregated Municipal Solid Waste. *J. Comput. Inf. Sci. Eng.* **2020**, *21*, 014502. [\[CrossRef\]](#)
22. Tachikake, H.; Watanabe, W. A Learning-based Robotic Bin-picking with Flexibly Customizable Grasping Conditions. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 9040–9047. [\[CrossRef\]](#)
23. Matsumura, R.; Domae, Y.; Wan, W.; Harada, K. Learning Based Robotic Bin-picking for Potentially Tangled Objects. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 7990–7997. [\[CrossRef\]](#)
24. Wang, H.; Niu, W.; Zhuang, C. GraNet: A Multi-Level Graph Network for 6-DoF Grasp Pose Generation in Cluttered Scenes. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 937–943. [\[CrossRef\]](#)
25. Liu, C.L.; Huang, T.H. Dynamic Job-Shop Scheduling Problems Using Graph Neural Network and Deep Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 6836–6848. [\[CrossRef\]](#)
26. Pistilli, F.; Averta, G. Graph Learning in Robotics: A Survey. *IEEE Access* **2023**, *11*, 112664–112681. [\[CrossRef\]](#)
27. Kokkalis, K.; Konstantinidis, F.K.; Tsimiklis, G.; Amditis, A. Optimization Framework of a Robotic Pick and Place System for Waste Sorting. In *European Robotics Forum*; Springer: Cham, Switzerland, 2024; pp. 246–251. [\[CrossRef\]](#)
28. Kokkalis, K.; Konstantinidis, F.K.; Tsimiklis, G.; Amditis, A. A Flexible Robotic-Based Architecture for Cyber-Physical Sorting Systems in Waste Management Industry. In *European Robotics Forum*; Springer: Cham, Switzerland, 2024; pp. 410–415. [\[CrossRef\]](#)
29. Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C.; Woodall, W. Robot Operating System 2: Design, architecture, and uses in the wild. *Sci. Robot.* **2022**, *7*, eabm6074. [\[CrossRef\]](#)
30. Sucan, I.A.; Chitta, S. MoveIt. 2025. Available online: <https://moveit.ros.org> (accessed on 30 June 2025).
31. Konstantinidis, F.K.; Sifnaios, S.; Arvanitakis, G.; Tsimiklis, G.; Mouroutsos, S.G.; Amditis, A.; Gasteratos, A. Multi-modal sorting in plastic and wood waste streams. *Resour. Conserv. Recycl.* **2023**, *199*, 107244. [\[CrossRef\]](#)
32. Tsai, R.; Lenz, R. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. Robot. Autom.* **1989**, *5*, 345–358. [\[CrossRef\]](#)
33. Yu, C.; Liu, X.J.; Qiao, F.; Xie, F. Multi-robot coordination for high-speed pick-and-place tasks. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macao, China, 5–8 December 2017; pp. 1743–1750. [\[CrossRef\]](#)
34. Huang, Y.; Chiba, R.; Arai, T.; Ueyama, T.; Ota, J. Robust multi-robot coordination in pick-and-place tasks based on part-dispatching rules. *Robot. Auton. Syst.* **2015**, *64*, 70–83. [\[CrossRef\]](#)
35. Murphy, R. *Introduction to AI Robotics*, 2nd ed.; Intelligent Robotics and Autonomous Agents Series; MIT Press: Cambridge, MA, USA, 2019.
36. Kokkalis, K.; Konstantinidis, F.K.; Tsimiklis, G.; Amditis, A.; Frangos, P. Modelling and Simulation of Industrial Delta Robots in ROS2 and Gazebo. In *European Robotics Forum*; Huber, M.; Verl, A.; Kraus, W., Eds.; Springer: Cham, Switzerland, 2025; pp. 81–87. [\[CrossRef\]](#)
37. Dave Coleman. About 85% of all Robotic Arms Now Have ROS 2 Drivers. 2025. Available online: <https://discourse.openrobotics.org/t/about-85-of-all-robotic-arms-now-have-ros-2-drivers/43835> (accessed on 28 June 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.