Robotics and Computer–Integrated Manufacturing 88 (2024) 102734

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Robust dynamic robot scheduling for collaborating with humans in manufacturing operations

Gilde Vanel Tchane Djogdom [a,b,*], Ramy Meziane [a,b], Martin J.-D. Otis [a]

[a] Laboratory of Automation and Robotic interaction (LAR.i), Department of Applied Sciences, Université du Québec à Chicoutimi (UQAC), 555 Boulevard de l'Université, Chicoutimi, QC G7H 2B1, Canada

[b] ITMI (Technological institute of industrial maintenance), Sept-iles College, 175 Rue de la Vérendrye, Sept-Îles, QC G4R 5B7, Canada

ARTICLE INFO

Keywords:
Human-robot collaboration
Travelling salesman problem
Knapsack problem
Time related human variability

ABSTRACT

The advent of collaborative robotics in industry has created a closer collaboration between humans and robots. This has led to the need to optimally schedule human and robot tasks to be robust enough to handle variability induced by time-related operator errors caused by the inability to accurately forecast the stochastic nature of human behavior. This article proposes an explicit scheme for tackling time-related variability in human tasks online in applications where humans intervene at a given time in the collaborative workspace. The planning problem is reformulated as a Travelling Salesman Problem combined with a 0/1-Knapsack Problem in order to actively define robot behavior when there is an unmodelled shift in the human execution time sequence. The method uses a two-level adaptation scheme. The first one (offline) inputs the predicted human behaviour in terms of time required for different activities at each work cycle, and then computes an overall task schedule to minimize the robot's operation time and idle time. The second one (online) involves the real-time detection of the human's timing to either stop the prescribed plan or enhance it in order to minimize robot and human idle times, thereby optimizing the sense of ease and fluency in the interaction. The system is simulated in different scenarios where the human predicted time is set to be wrong, and thus the system needs to account for such variation. The effect of the human predicted time on the task schedule is presented and helps to demonstrate the effectiveness of the proposed approach in dealing with human variability without prior modeling knowledge of the human task time distribution.

1. Introduction

The advent of collaborative robots in industry has naturally raised the question of how to effectively use both partners (robot and human) in order to optimize production time while preventing the development of musculoskeletal disorders. Many attempts have been formulated to assess such a question. Among them, some scientists tackled the problem in a static way by defining an optimal or near-optimal allocation of tasks offline and scheduling between a human partner and a collaborative robot based on technological, physical, and skill constraints [1,2]. However, in such an approach, the production environment is static, which in fact does not correspond to reality [3]. Another approach considers the environment as being dynamic while humans are considered as fully controllable entities [4–6]. Based on this assumption, various solutions that can deal with unforeseen events and changes in actor capabilities or cell states are being investigated. Among them are online scheduling and task assignment strategies that explicitly consider factors such as, the human's accumulated fatigue over time [4], the human's experience with the task [2], the trust between the human and the robot partner [5,7], or even the physical ergonomics of the human partner [6], etc. However, considering the human as a fully controllable entity restricts its adaptive properties which were initially sought as an asset for human-robot collaboration [8,9].

Another scheduling approach aims to preserve the human adaptive property for rapid reconfiguration. Here, the robot partner is viewed as a follower that must adapt to the perceived behavior of the human operator. The behavior can be explicitly expressed through commands [9] or can be predicted through analysis of the human operator's movement during the collaborative process [8,10] Based on this approach, the robot selects the tasks that correspond to the human's perceived needs and preferences [11]. This reactive scheduling generally considers ergonomics factors of the human partner and consider either static [12] or

* Corresponding author.
E-mail address: gilde-vanel.tchane-djogdom1@uqac.ca (G.V. Tchane Djogdom).

https://doi.org/10.1016/j.rcim.2024.102734
Received 9 May 2023; Received in revised form 5 November 2023; Accepted 28 January 2024
Available online 9 February 2024

mobile robots [13]. Moreover, this approach aims to consider the human as the main element in the interaction and is not well suited to cases where the operator is only used at a given time in the interaction. Moreover, when the complexity of assembly or disassembly tasks increases, such a structure tends to show its limits [14]. The three solutions mentioned above consider the human's behavior as purely normal or even deterministic in the collaborative scenario, which is not true in real life applications [15–17]. A scheduling approach is formulated to enable the scheduler to manage unexpected event related to human behavior (omitting an action, repeating an action etc.) or to robot failure, in order to address potential errors during collaboration [15,16]. It is commonly assumed that humans have a higher rate of unforeseen failures than their robot partners in collaborative work, due to their ability to make spatial, temporal or even contextual errors [17].

In the literature, there are common scheduling approaches that deal with spatial or contextual human errors with the goal of ensuring human security [15] and maintaining productivity continuity through dynamic rescheduling [16,18,19]. More recently, some works have focused specifically on the temporal variability due to inconsistencies of the partners (both human and robots) in performing their different tasks. Some of the research works use an offline stochastic approach in their formulation to account for the temporal deviations of the different partners by explicitly including a time-dependant deviation model [3]. However, such an approach assumes that the temporal variation patterns of the partners are fixed and known in advance. In a pilot study, [20] show that the limitation of such an approach is that the scheduler is not able to effectively accommodate all possible temporal deviations. They propose a stochastic temporal variation pattern that can be updated at each cycle time. However, these approaches assume state that the scheduler modifies the human's task online by reassigning his task. For configurations where changes in robot mode can occur due to the presence or absence of the human in the collaborative cell, and by considering the human tasks as known and fixed, no approach is formulated for the scheduling structure necessary to accommodate any temporal variability of the human partner based solely on the rescheduling of the robot task.

This article is mainly interested in cases where the human is considered to be completely uncontrollable by the robotic planner which in this case is used as a subcomponent of the global planner (factory management). In this type of application, the tasks of the human are predefined with precise temporal constraints of intervention in the workspace and for which the temporal variations are random.

Such a problem can be reformulated as a combination of the Traveling Salesman Problem (TSP) and the Knapsack Problem (KP). In such an approach, the TSP problem formulation allows to compute the optimal path production sequence passing through all subtasks of interest while respecting technological constraints. In the literature, this TSP formulation has already been used within the robotic scheduling approach to determine the autonomous optimal disassembly sequence of a product [21–23]. However, such an approach cannot be applied when considering the task time related variability of the resources in a human-robot collaborative (HRC) scenario. In order to enable the scheduler to deal with time-related uncertainties, this formulation can be enriched with a Knapsack (KP) type approach to find the sequence of actions that satisfies a given cost criterion [24]. KP problem formulations are widely used in the literature for the U-line balancing problem to obtain the maximum breakpoint for the piecewise approximation of the standard deviation of the station completion time where the task time variability is considered as stochastic (independent, normally distributed variable with known means and variances). In fact, this cannot be used for human time-dependent deviations [20]. The question of defining which robot behavior results from the human partner's time-related deviations is an open issue and from the author's current knowledge, the explicit implementation of these different problem formulations for human-robot scheduling in the HRC scenario has not yet been explored.

The contributions of this article are as follows :

1) The proposal of a planning formulation in which a robot can ostentatiously modify its task scheduling sequence based on the human's predictive temporal information.
2) The reformulation of the Human-Robot scheduling problem as a set of standard traveling salesman and 0/1-knapsack problems.
3) The formulation of an additional online correction approach to account for prediction errors that may occur in real operation with the goal of minimizing the idle time of the human and the robot, as well as the production sequence.

This research work mainly aims to reschedule real-time robot tasks only by explicitly considering human temporal variability in a proactive and reactive manner reformulated as a combination of combinatorial optimization problems such as TSP and KP. As the number of research works in this area is increasing, Section 2 reviews the related works to contextualize the contribution of this research work. Section 3 presents the proposed approach used which is a predictive reactive approach for online HRC scheduling based on sub-problem formulations such as TSP and KP. Section 4 presents the case application used to demonstrate the effectiveness of the proposed approach. Section 5 presents the results and discussion. Section 6 presents the limitation of the study and Section 7 presents the conclusion of the study and future work.

2. Related works

A brief review of existing studies on scheduling methods for dealing with task time variability is presented here. Studies on the use of AOG (AND & OR Graph) as a task modeling approach are explored with the case study of formulating scheduling problems such as TSP (Traveling Salesman Problem) and KP (Knapsack Problem).

2.1. Scheduling approach based task time variability

The question of how to account for temporal considerations in the planning of human and robot tasks has been addressed in different ways in the literature. Some authors [25] studied a planning structure in which the amount of time needed for each agent to complete its tasks is considered constant. Others extended the analysis by formulating a bounded range of variations in tasks times in [26–28]. However, regardless of which of these approaches is used, such a structure proves to be inefficient in capturing the variability inherent in the different agents, especially that induced by the human partner. In their studies, [10] and [29] proposed to model humans as stochastic entities using a predefined distribution model to account for task time variations. A reactive adaptive strategy is then proposed to adapt to the different scenarios based on the task with the fastest execution time. However, such a reactive approach is inadequate in its ability to account for the temporal deviations caused by the inconsistency of humans in performing their tasks. In fact, even with a reactive approach, it is not realistic to consider the distribution of the human task time as fixed and stationary. A planning structure based on a bipartite approach is presented in [3]. On the one hand, it combines an offline planning sequence in which the temporal variation induced at the level of the different agents follows a Gaussian distribution known a priori. In this phase, a multi-agent approach is implemented to determine a priori an optimal planning sequence. Subsequently, an additional online phase is indexed, which consists in updating the actual behavior during online operation. This online sequence uses a set of pre-defined heuristic rules to handle any deviations from tasks and plans that become impractical. However, as pointed out in [20], this strategy cannot derive an optimal plan in cases where the temporal models of the different agents are not stationary. In [20], this theme is indexed by proposing an approach of online adaptation of the temporal distribution model of the tasks of the different agents via a receding horizon. However, these works consider the human as a controllable or partially controllable entity whose task dispositions can be modified during operation. In [30], an approach is proposed where safety parameters for collaborative tasks are considered as a source of robot time variability corresponding to human task time variability. He formulates a rescheduling approach as a mixed-integer problem to find the optimal task allocation in real time. More recently, [31,32] propose structures that are not based on a temporal distribution model of the tasks of the different agents, but based on how to recover and predict the operation times. As a predictive approach, the probable future temporal structures can be accommodated online. However, these works consider the possibility of modifying the human behavior to find a collective synergy between all the agents in order to minimize the idle time of the resources. Moreover, according to [20], the scheduling approach is purely based on the accuracy of the task duration prediction. In general, approaches in which structures consider the temporal variability of the human use either conservative approach [19] (i.e., the robot stops while waiting for the human to be available), reactive approaches [10,29] (i.e., at each moment, the least costly task that can be performed by the robot is executed while waiting for the human operator to be available), reactive approaches based on heuristic rules, or a proactive structure [31,32], where absolute confidence in the prediction element is used to predict the remaining time for completing the tasks and to compute the sequence of actions that can be performed in this interval. However, these approaches give absolute confidence to the prediction element without considering the effect of prediction error. Table 1 presents a summary of the different scheduling approaches based on their ability to handle the time variability of human tasks.

In, it can be seen that none of the approaches considers the fact that human task durations may follow a non-fixed, non-stationary distribution, together with the ability to update the distribution model. Also, none of them consider the temporal predictions based on online estimation of task durations, rather they consider the human as a controllable agent in a predictive-reactive manner. The proposed method is based on two-step approach. First, at each cycle time of work, a prediction of human task availability or duration is computed, then an optimal plan based on sub-problem reformulation such as TSP & KP solving problem is formulated. Then in real time, based on the actual perceived human task time, the plan is modified in a reactive manner. This approach can only be applied for robot task where the human is considered as an uncontrollable agent. Moreover, it does not consider the human task time distribution as fixed and static. The next section reviews the sub-problem reformulation in task planning application in HRC (Human Robot Collaboration).

2.2. TSP and KP Sub problem reformulation in HRC

The issue of planning has long focused on finding the optimal path to minimize a given criterion. This problem involves minimizing either one or multiple objectives and, in some cases, can be reduced to combinatorial optimization type problems, such as the bin packing problem, vehicle routing problem, etc. In the field of autonomous robotics, the traveling salesman problem (TSP) formulation is frequently employed to determine the optimal task sequence for disassembly or assembly offline, the [21,33–38]. The TSP is traditionally defined as the problem of visiting a given set of cities once from the city or point of origin and returning to it [22]. In [33], the assembly problem of a Gear box is defined as a modified TSP formulation eliminating the requirement to return to the city of origin. This problem formulation often requires the problem to be defined as a behavior tree or AND/OR graph modeling approach. A novel TSP formulation for disassembling print circuit boards is proposed in [34]. The TSP formulation, utilized in mobile robotics, has been utilized for both path reconfiguration [21] or inspection [37]. In the realm of human- robot collaborative (HRC) scenarios, the TSP formulation is often utilized offline to achieve optimal or near-optimal task scheduling [39]. A proposed formulation in [38] showcases human and robot collaboration to accomplish individual tasks (human or robot) or joint tasks (human and robot). A rescheduling strategy relies on Bayesian inference to predict the human next actions at each time point. This information is then utilized in a multi-agent formulation of TSP that is solved online to reschedule the robot tasks. However, addressing the task time variations combined with changes in the human-robot interaction mode presents a challenging problem for any TSP-based solution.

On the other hand, the Knapsack (KP) problem formulation has been used in the field of human-robot collaboration for optimal resource allocation [40–42], robot path planning [43], or task scheduling [44,45]. Historically, the Knapsack model arises from the problem of selecting among a set $I = \{1,\ldots,N\}$ of possibly identical items, those to be packed into containers of limited capacity in order to maximize some utility value [46]. To address the issue of optimal resource allocation, [40] proposes the use of the KP problem for resource allocation based on collaborative multi-agent systems. Moreover, in [41], a multi-knapsack problem for path welding based on multi-robot application is presented. In their study, [42] present a formulation of the knapsack problem to efficiently manage the organization of production resources and ensure compliance with time and capacity constraints in a flexible manufacturing setup. In the field of robot path planning, [43] reformulated the path planning problem for distributed robot beamforming under motion energy constraints as a KP problem. In such a formulation, the robot is instructed to move to find locations that satisfy the given quality requirement while minimizing the total energy consumption. For task scheduling approaches, [44] formulated a multi-station test scheduling optimization method for an industrial robot servo system as a modified knapsack problem formulation. In [45], a knapsack formulation is used to solve the scheduling approach of a single machine with chain structure precedence constraints and separation time windows. The goal is to reduce the time taken to complete a task. In [31], the knapsack problem is formulated to solve the multi-robot task assignment algorithms, considering practical constraints such as task deadlines and limited battery life of robots. However, in the field of human-robot collaborative scenario, [47] proposes a multi-choice multi-dimensional knapsack formulation problem for robot site inspection, subject to constraints and human guidance. The goal is to optimize motion, sensing and human queries.

However, from the author's point of view, the Knapsack problem formulation has not yet been used for task time variation and online robot task planning in human-robot collaboration . The next section presents the proposed scheduling approach which is the coupling of TSP and KP problem formulations.

3. Proposed optimisation system

First, the task problem is defined in Section 3.1 to describe the different subsets of tasks used in the proposed approach. Then, in Section 3.2 the offline task planning based on the TSP and KP formulations is presented. Finally, Section 3.3 presents the online adaptation of the proposed plan.

3.1. Task problem definition

The human-robot interaction in the same workspace requires the definition of the types of tasks to be performed. It is generally assumed that in such configuration there are two main categories of tasks, which are:

➢ Parallel tasks, which refer to tasks that can be performed simultaneously by the two agents (human and robot) in the same workspace. These tasks may and may not be joint tasks.
➢ Non-parallel tasks are those that cannot be performed in parallel due to safety or ergonomic constraints.

Table 1
Summary of scheduling approaches based on their ability to handle tasks time variability.

Article	Human Task time				Human as non controllable agent	Scheduling approach			Method	Comment
	Not Fixe distribution	Not Stationary distribution	Adaptive / learning distribution	Time estimation based on online prediction		Offline	Online			
							Reactive	Predictive		
Proposed approach	Y	Y	Y	Y	Y	Y	Y	Y	DP	Uses of Dynamic Programming (DP)to solve sub-problems formulated as TSP & KP formulation
[19]	Y	Y	Y	Y	N	N	N	Y	A*	Time variability is encapsulated as a cost value of the task and solved with an A* at each cycle to find a suboptimal path
[20]	Y	Y	Y	N	Y	N	N	Y	TPN	Adjustment of online task time distribution using time Petri nets as a model and a receding horizon
[25]	N	N	N	N	N	Y	N	N	HNP	The problem is formulated as a mixed integer linear problem and solved using a hybrid nested partition algorithm (HNP)
[28]	N	N	N	N	N/A	Y	N	N	GA	Use of a fixed stationary time distribution for the task time. The scheduling problem is modeled in the form of GSPN (Dual generalized petri nets) and solved offline as a multi-objective optimization problem (MOOP)
[27]	N	N	N	N	N	Y	Y	N	GA	Use of genetic algorithm to find suboptimal task allocation. Reactive approach is based on selecting the task with the earliest finish time as the next to be executed for human
[26]	N	N	N	N	N	Y	N	N	MILP solver	Mixed integer linear problem formulation to find suboptimal scheduling
[10]	Y	Y	Y	Y	Y	N	Y	N	If condition	Reactive behavior based human synchronization for delivery parts (when to start or not)
[29]	Y	Y	Y	Y	Y	N	Y	N	If condition	Reactive behavior for triggering quality check based on task time duration comparison
[3]	N	N	N	N	N/A	Y	Y	Y	If then conditions	Create a pre-defined schedule based on fixed and static Gaussian time distribution and repair online using reactive based heuristic rule approaches (machine doesn't account for human in the interaction loop)
[30]	Y	Y	Y	Y	N	Y	Y	N	MILP	Human and robot task time variability: Compute offline nominal scheduling and modify online this approach based on perceived real task durations formulated as a MILP (Mixed Integer Linear Programming) problem
[31]	N	N	N	N	N	Y	N	Y	ACA	Explicit formulation of task time variability in offline programming and optimization solutions using the Agglomerative Clustering Algorithm (ACA)
[32]	Y	Y	Y	N	N	Y	Y	N	TPN (time petri nets)	Switch between precomputed plan based on remaining task completion time

From these two main categories, four (4) subtask categories can be defined as follows:

- ➢ Type 1 (T1), which refers to human task only.
- ➢ Type 2 (T2), which refers to robot task only.
- ➢ Type 3 (T3), which refers to tasks that can be performed by either à human or a robot, but with the constraint that only one resource is assigned at a time.
- ➢ Type 4 (T4), which refers to a joint human and robot task.

Fig. 1 presents an overview of task classification in a human–robot collaborative scenario.

In the proposed scenario, the type 3 task (T3) is not considered, since the human task set is assumed to be known a priori and cannot be changed during the interaction. Thus, only T2 and possibly T4 type tasks are indexed in this article. In addition, an assumption has been made about the parallel and non-parallel state of T2 tasks, as some may be performed in human presence or not due to security requirements.

Furthermore, in this problem formulation, a job is considered as a set of tasks or operations to be performed by the robot partner to assemble or disassemble a product made of N parts. A task is defined as a fixed sequence of actions that transforms a stable intermediate state into another one. Moreover, this formulation assumes that tasks are not preemptible.

The formulation of the scheduling problem aims at finding the optimal sequence that allows to reach a given predefined operational state while minimizing a cost criterion. More specifically, this objective can be defined as the desire to find an optimal production sequence while minimizing the downtime of the robot and the human in the workspace as well as the total production cost. To achieve this goal, the technological constraints between the different tasks must be taken into account. Therefore, it is necessary to use a modeling tool that can represent the state of the cell as well as the possible evolutions. In general, to have a complete and compact representation of the production sequences, the AOG (AND/OR GRAPH) is usually used [32]. It is a hypergraph H = (N, E), where N is the set of nodes and E is the set of Hyper (Edges) of the graph. In such a representation, the nodes refer to the state of the production while the edge refers to the precedence relations to go from the current state to the next state. It also refers to the work in process or one that is required to go from one state to another. Each edge is associated with a cost. All arrivals edge to a node can infer either an AND relationship if there is an arc related to them or an OR relationship otherwise. Fig. 2 below shows an AOG model.

In the model shown in Fig. 2, S1, S2, S3 and S4 tasks require S0 task to be completed before they can be performed. In addition, the S5 task with the joint arc implies that S1, S2, S3 and S4 must be completed before the S5 task can be performed. The value associated with the cost of going from task S0 to task S2 is defined as 2. For task S9, it can be seen that task S9 can be performed if S8 has been done or S6 and S7 have been done.

In addition to the above formulation of the AOG graph, the parallel and non-parallel state of the robot task (T2) is considered in the problem formulation. Thus, the scheduling problem used in this article requires the formulation of four subtask categories:

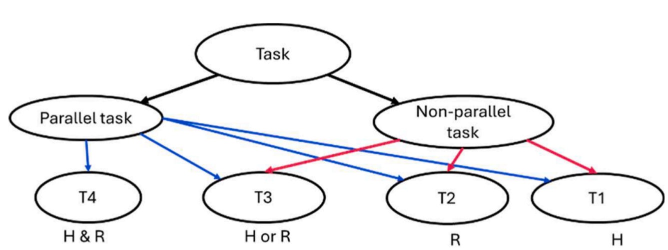

Fig. 1. Tasks in H – R collaboration scenario.

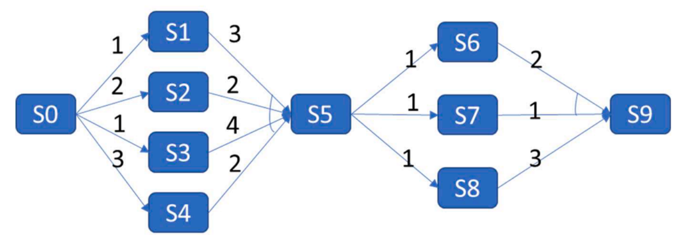

Fig. 2. AOG model.

- ➢ Ts1 refers to all tasks that must be computed before the human can enter the collaborative workspace. They are all T2 tasks and are non-parallel. Once completed, humans can enter the collaborative workspace at any time.
- ➢ Ts2 refers to all tasks that can be added to the robot's execution schedule before actual human intervention. This set of tasks can include non-parallel or parallel T2 tasks, since it is assumed that all tasks considered in this step have the same precedence characteristics but differ based on their cost and parallel or non-parallel state.
- ➢ Ts3 refers to all tasks which are to be performed by the robot partner in the presence of the human operator. It can be a joint task T4 or only a parallel task T2.
- ➢ Ts4 refers to tasks which need to be conducted after the human partner has left the collaborative space.

The overall scheduling approaches used in this formulation imply the use of proactive and reactive scheduling scenarios. Fig. 3 below shows a brief overview of the proposed approach.

The next step presents the proactive algorithm for computing the optimal robot task scheduling based on the predicted human execution time.

3.2. *Proactive scheduling approach*

The proactive algorithm aims to define the optimal set of tasks to be performed by the robot based on the human's predicted execution time. The algorithm formulation is shown in Fig. 4 below.

The algorithm shown in Fig. 4 considers two main characteristics for one sequence of human interventions in the collaborative workspace: First the predicted time availability of the human to enter the collaborative workspace and second the predicted duration of human intervention in the collaborative workspace. These two characteristics along with the product assembly or disassembly model are used as input to the scheduler. Thus, a set of four steps is performed to derive the optimal robot scheduling plan:

- ➢ Step1: The goal of this step is to define the optimal scheduling of the **Ts1** robot task. As defined earlier, it refers to tasks that need to be performed before the human partner is allowed to enter the workspace. In the assembly or disassembly sequence, such tasks induce the need to formulate the scheduling problem as the Traveling Salesman Problem without the requirement to go back to the initial node . The TSP aims to find the optimal path considering all the nodes. For example, there may be some cases where, as shown in Fig. 2, in order to do S5, it is necessary to do S0, S1, S2, S3 and S4. The result of this step is the optimal scheduling of Ts1 tasks based on minimizing the total cost of production. These tasks are represented in green color.
- ➢ Step 2: The goal of this step is to add robot tasks to the collaborative workspace based on the predicted human availability. If the human is ready for the task at the right time, this step may not be necessary. This step is taken when the predicted human availability is greater than the lowest task cost of the Ts2 set of tasks. At this step, a 0/1-Knapsack Problem (KP) is formulated based on the cost or weight

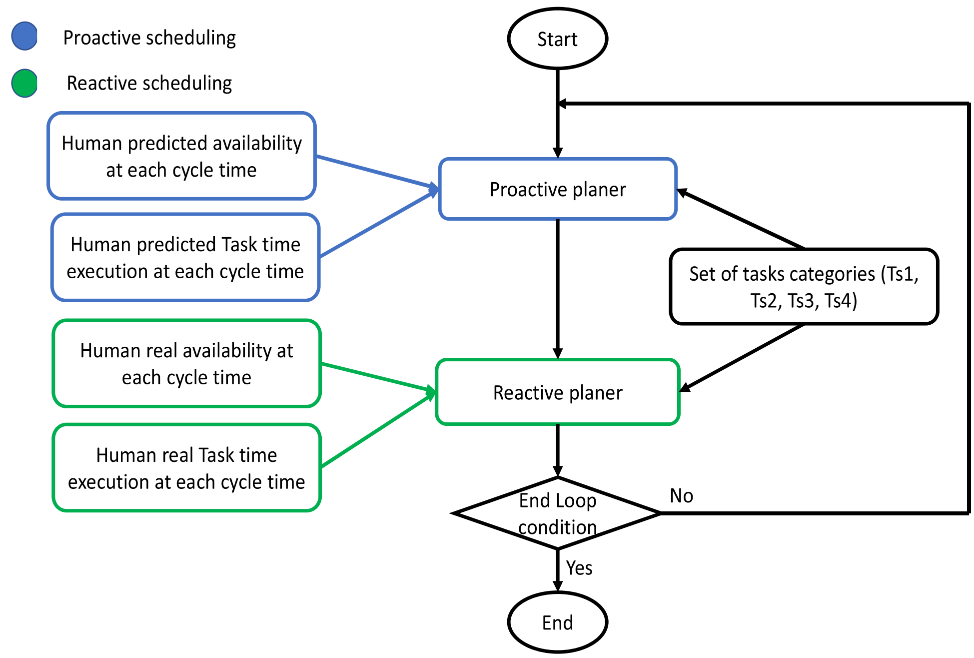

Fig. 3. Overall tasks planning algorithm.

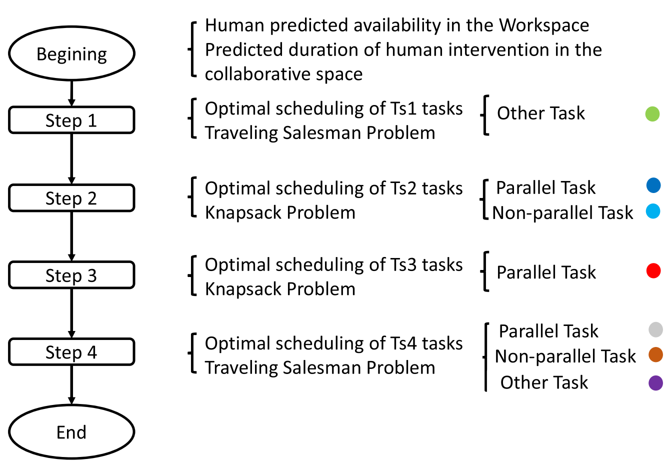

Fig. 4. Proactive scheduling algorithm.

of each task and its values. In this formulation, the task values refer to their parallel or non-parallel state. In fact, in this step, the problem is formulated to maximize the robot task while considering the human delay by prioritizing the non-parallel tasks, since they cannot be done at the same time with the human operator. However, when all non-parallel tasks have been assigned, the parallel tasks come next. The result of this step is the optimal scheduling of Ts2 tasks with the goal of maximizing the number of tasks to be added according to human availability. This reduces the idle time of the robot caused by the variability of the human time. Parallel tasks T2P belonging to this step are represented in light blue and non-parallel tasks (T2NP) are represented in dark blue.

➢ Step 3: The goal of this step is to determine the optimal scheduling of robot tasks in collaboration with the human partner. These tasks refer to Ts3 types. The 0/1-knapsack Problem is formulated as an optimization problem to prioritize joint human-robot actions over cooperative ones while considering the weight of production cost or time and the values and ensuring that all precedence constraints are satisfied. Under this assumption, the knapsack problem concerns T2 parallel tasks and T4 tasks. The result of this step is the optimal scheduling of the set of robot actions (Ts3) to maximize the duration of human intervention and thus minimize the robot idle time and the overall productivity cost. For illustration purposes, the parallel tasks T2P belonging to this step are represented in red.

➢ Step 4: The goal of this step is to define the optimal arrangement of the set of tasks which needs to be performed after the human operator leaves the collaborative workspace. Such a task is defined by prioritizing the T2 parallel tasks that have not been integrated into the optimal Ts3 task set. Furthermore, the T2 non-parallel task that has not been integrated into the Ts2 task set is integrated. Then, the previously defined Ts4 task that has reached the desired goal is reformulated as a Traveling Salesman Problem to find the optimal path. The result of this step is the optimal scheduling of the new Ts4 task set, which leads to the production goal. For illustration purposes, parallel tasks T2P belonging to this step are represented in light grey, non-parallel tasks (T2NP) in dark orange and other tasks in purple.

Once defined, the planner must react to any deviations in the pre-computed path.

3.3. Reactive scheduling approach

When implementing the proactive approach, there may be a scenario where the predicted human behavior is not true, so the scheduler needs to account for such case. The approach consists of updating the robot scheduler in three phases. First before detecting the human arrival, if all the original **Ts2** tasks have already been executed it means that the human availability time has been underestimated. Then, at each time step, the scheduler sends to the robot the lowest task among all the available ones by prioritizing non-parallel **T2** task over the parallel one. On the other hand, if human availability time has been overestimated, then the scheduler must interrupt the other **Ts2** tasks to minimize the human idle time. Then, the unexecuted **Ts2** tasks will be sent for backup in the next step. The second update occurs when the human is being too slow or too fast according to the predicted pace. Then, the scheduler must either stop the execute of the **Ts3** task or supplement it at each time step with the lowest available **T4** or **T2** parallel task that has not yet been executed. The last update occurs when the human has left the collaborative space. Then, the scheduler must modify the precomputed **Ts4** set of tasks to account for the tasks that have been executed with previously unexecuted tasks. The algorithm below presents an overview of the proposed rescheduling algorithm.

Algorithm: Reactive scheduling

```
Input: Ts1, Ts2, Ts3, Ts4, T2P, T2NP
Output: planning action of the robotic partner
#T2NP refers to non-parallel T2 tasks that have not been integrated in Ts1 and Ts2
  solutions
#T2P refers to parallel T2 tasks that have not been integrated in Ts2 and Ts3 solutions
        i = 0, w = 0, j = 0, k = 0
        for i = 0 to len (Ts1) do:
                Execute (Ts1(i))
        end for
# Detecthum () # Launch real time detection of human in the collaborative cell
  (Boolean)
      While (Detecthum () < 1) do: # human not detected.
          if i <= len (Ts2) then
              i = i + 1
          execute (Ts2(i))
      end if
              if i > len (Ts2) AND j <= len(T2NP) AND len(T2NP) > 0 then
              j = j + 1
                    execute(T2NP(j))
        end if
        if i > len (Ts2) AND j > len(T2NP) AND k <= len(T2P) AND len(T2P) >
  0 then
              k = k + 1
              execute(T2P(k))
        end if
      end while
# Save all Ts2 files that have not been executed along with T2P or T2NP that have not
  previously been integrated in Ts2
upload (T2P, T2NP) # Remove T2P and T2NP that were executed in the previous loop
  and add the remaining actions of Ts2 that were not executed in the previous while
  loop.
# human has entered the collaborative workspace
# Detecthum () # Launch real time detection of human in the collaborative cell
  (Boolean)
      i = 0, j = 0
      while (Detecthum () > 0) do:                                  #human is
  still in the loop.
          if i <= len (Ts3) then
          i = i + 1
        execute (Ts3(i))
        end if
            if i > len (Ts3) AND j <= len(T2P) then
        j = j + 1
      execute(T2P(j))
      end if
      end while
upload (Ts4) # Rescaling of Ts4 task based on modified Ts2 and Ts3 set of tasks.
for i = 0 to len (Ts4) do:
execute (Ts4(i))
end for
### Fin Algorithm.
```

In Section 4, a case study on the disassembly of a 2011 Nissan Leaf battery Pack in a human-robot collaboration is presented to evaluate the impact of such an approach and the need to proactively consider the time variability associated with the human operator's interactions with its collaborative partner. .

4. Case study

The case study concerns the extraction of Nissan Leaf battery cells based on collaborative disassembly for recycling purposes. In this process, humans are involved in external tasks such as the assembly of the power source or operations related to other products. However, since the battery model is so complex that, the robot alone cannot perform the disassembly tasks, a collaborative scenario involving human and robot is defined. The robot is assigned with the task of unscrewing, transporting and storing the components. The robot starts the disassembly process in an autonomous mode (the human is not in the collaborative cell).

For complex tasks, human intervention in a collaborative workspace is necessary to remove connection wires, cables, or harnesses since machine vision alone is insufficient. The robot must switch to collaborative mode, allowing for parallel work between the human and the robot. After the removal of wires and detection of the human departure from the collaborative cell detected, the robot's path is cleared to proceed to other tasks in an autonomous mode.

The installation's framework is created within RoboDK, a software for robot programming and simulation. The planning was developed using python. Fig. 5 provides an overview of the installation in the RoboDK environment, showing the battery in the workspace (the table). Moreover, the figure also displays the source to be mounted which is also part of another human-related tasks. In this figure, the human is located in the collaborative cell. Consequently, the robot must switch to a collaborative mode to enable the execution of parallel tasks in the presence of the human. This case study involves a shift from a solely autonomous mode without regard for human presence to a collaborative mode.

Fig. 6 provides an overview of the Nissan Leaf battery pack in RoboDK. Blocks 1, 2 and 3 indicate the cells that need to be dismantled, while the white eclipse indicates the wire arrangement that the human need to remove.

To address the optimal scheduling challenge of accounting for technological constraints across disassembly parts and the time-related variability introduced by the human partner during the disassembly process, we propose an AOG model as depicted in Fig. 7. The model outlines the necessary tasks to be performed by the robotic partner, with interrupted squares representing the battery sub-blocks and the largest block signifying the Nissan Leaf battery. The first block (1) details the steps for removing the battery cover, while the second block (2) outlines the process for removing the BMS (Battery management system). The top block (3) provides instructions on removing the cells located on the side of the robot. It is important to note that this block only covers robot's tasks that be performed in parallel with human intervention for safety reasons. The lowermost block, numbered 4, outlines the necessary steps for the removal of cells that are situated at the closest possible distance to the operator. These tasks must be done in the absence of a human partner. The subsequent sub-block, numbered 5, entails the tasks involved in extracting the cells in the battery compartment at the rear.

Fig. 7 displays the steps to completely disassemble a Nissan Leaf battery pack. The precedence relationships between each node are presented in red lines. In addition, each node is connected to at least one other node by an edge, showing the cost of going from one node to the another. An additional task S7 is added to represent the human action in the loop and the need for the human operator to remove the harness (cables/wires) inside the battery pack to allow the robot to access the nodes S22 to S29 steps. Table 2 defines the required tasks to be performed, their precedence relationships, their costs and their labeling as either parallel or non-parallel task (T2P or T2NP) and their potential to be part of Ts1, Ts2, Ts3 and Ts4 possible tasks.

Each task is characterized by a given cost which is the time (in minutes) required to complete the task. In this formulation, it is assumed that the robot task time is set to a nominal value that does not change even in the presence of the human partner in the collaborative workspace. Tasks time are provided directly given from the RoboDK simulation software as they refer to robot time to compute a given task. Table 3 below presents the nominal tasks completion time.

5. Results and discussion

Three scheduling approaches are compared through online simulation in the RoboDK environment on a computer with an AMD Ryzen 5 3500 CPU, with Radeon Vega Mobile Gfx 2.10 GHz and 8Gb RAM. They are:

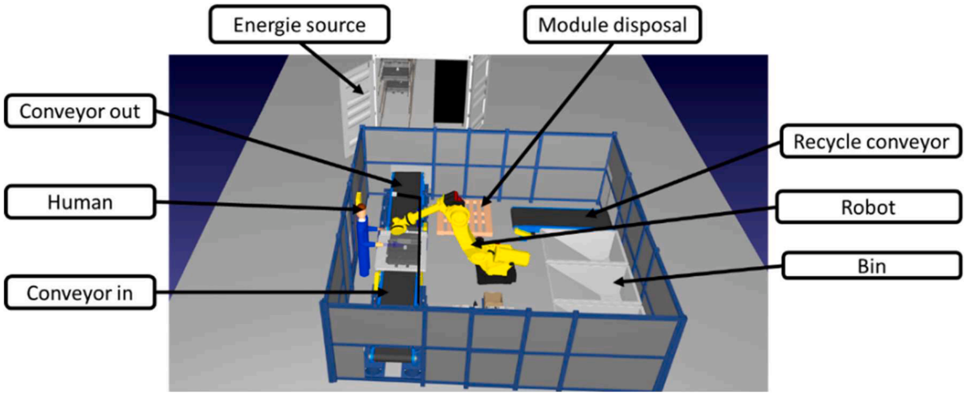

Fig. 5. RoboDK displays Nissan leaf battery disassembly.

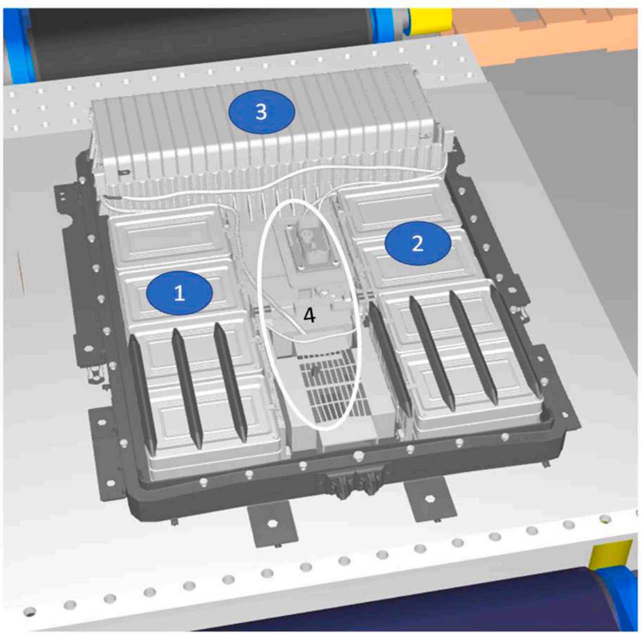

Fig. 6. Insight of the Nissan Leaf Pack in RoboDK.

- Proactive planner (P): It is based on the overconfidence of the human task time predictions. This resulting plan is executed in real time solely based on the predicted human task time without considering online monitoring of actual human task time.
- Reactive planner (R): It involves of determining the sequence of tasks to be performed at each time t based solely on technological and cost constraints. This means that for tasks with identical precedence constraints, will be executed based on the task with the lowest immediate cost, and so forth.
- Proactive – reactive planner (PR): It is the proposed approach outlined in this research work with the goal of merging the advantages of both prediction-based and reactive approaches.

Different use cases have been implemented. For analysis purpose, only one case will be displayed and it will be further enhanced with the observations derived from other uses cases.

Use Case: In this scenario, the predicted human availability time required to be ready to work is called as Tdp and is set to 12 minutes. The real human availability time is denoted as Tdr and is set to 14.813 minutes. The predicted human work time intervention in the collaborative space is denoted as Tip and is set to 3 minutes and the actual human work time intervention is denoted as Tir and is set to 1 minute. Fig. 8 below presents the scheduling results proposed by the three schedulers in minutes. In Figs. 8, 9, 10 below, the scheduling approach based respectively on a purely proactive planner (P), a purely reactive planner (R) and a Proactive-Reactive planner (PR) is presented. in the x axes, the task time are presented in minutes.

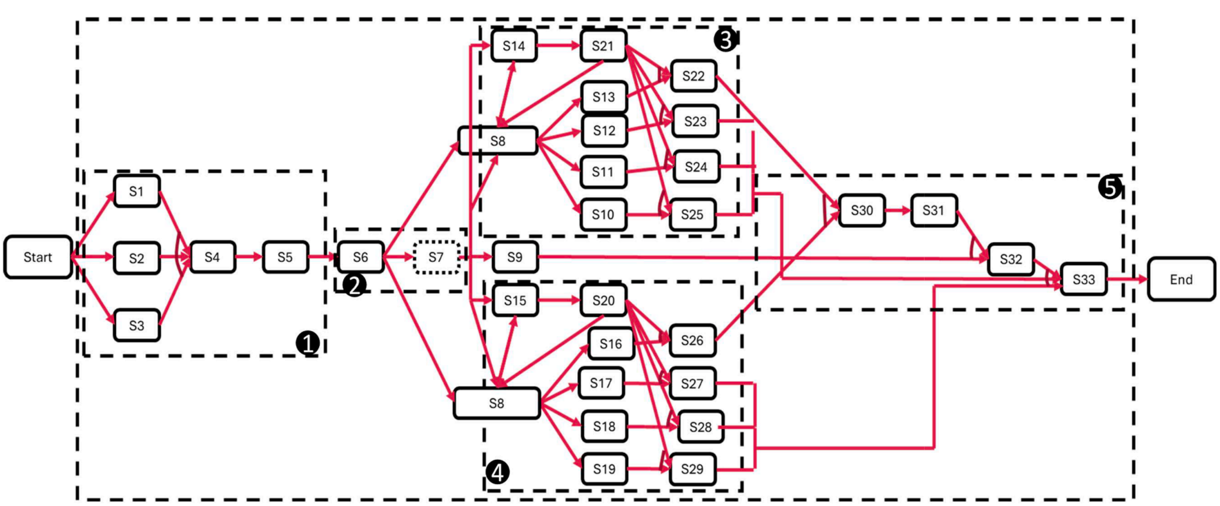

Fig. 7. AOG model for the 2011 Nissan Leaf battery disassembly plan.

Table 2
Tasks signification and categorisation.

Node's name	Signification	Prerequisite node	T2P	T2NP	Ts1	Ts2	Ts3	Ts4
S1	Type 1 screw removed	Start	☐	☐	☒	☐	☐	☐
S2	Type 2 screw removed	Start	☐	☐	☒	☐	☐	☐
S3	Type 3 screw removed	Start	☐	☐	☒	☐	☐	☐
S4	Intermediate plate removed	S1 & S2 & S3	☐	☐	☒	☐	☐	☐
S5	Cover moved	S4	☐	☐	☒	☐	☐	☐
S6	BMS unscrewed	S5	☐	☐	☒	☐	☐	☐
S7	Human intervention for wire and BMS removal	S6	☐	☐	☐	☐	☐	☐
S8	Type 1 tool changed	S6	☒	☐	☐	☒	☒	☒
S9	Rear support block removed	S6	☐	☐	☐	☐	☐	☒
S10	Right block support maintaining compartment 4 removed	S8	☒	☐	☐	☒	☒	☒
S11	Right block support maintaining compartment 3 removed	S8	☒	☐	☐	☒	☒	☒
S12	Right block support maintaining compartment 2 removed	S8	☒	☐	☐	☒	☒	☒
S13	Right block support maintaining compartment 1 removed	S8	☒	☐	☐	☒	☒	☒
S14	Right block electrical junction screw removed	S6	☐	☒	☐	☒	☐	☒
S15	Left block electrical junction screw removed	S6	☐	☒	☐	☒	☐	☒
S16	Left block support maintaining compartment 1 removed	S8	☐	☒	☐	☒	☐	☒
S17	Left block support maintaining compartment 2 removed	S8	☐	☒	☐	☒	☐	☒
S18	Left block support maintaining compartment 3 removed	S8	☐	☒	☐	☒	☐	☒
S19	Left block support maintaining compartment 4 removed	S8	☐	☒	☐	☒	☐	☒
S20	Left block electrical junction moved	S15	☐	☒	☐	☒	☐	☒
S21	Right block electrical junction moved	S14	☐	☒	☐	☒	☐	☒
S22	Right block cells compartment 1 removed	S7 & S13 & S21	☐	☐	☐	☐	☐	☒
S23	Right block cells compartment 2 removed	S7 & S12 & S21	☐	☐	☐	☐	☐	☒
S24	Right block cells compartment 3 removed	S7 & S11 & S21	☐	☐	☐	☐	☐	☒
S25	Right block cells compartment 4 removed	S7 & S10 & S21	☐	☐	☐	☐	☐	☒
S26	Left block cells compartment 1 removed	S7 & S16 & S20	☐	☐	☐	☐	☐	☒
S27	Left block cells compartment 2 removed	S7 & S17 & S20	☐	☐	☐	☐	☐	☒
S28	Left block cells compartment 3 removed	S7 & S18 & S20	☐	☐	☐	☐	☐	☒
S29	Left block cells compartment 4 removed	S7 & S19 & S20	☐	☐	☐	☐	☐	☒
S30	Rear block electrical junction screw removed	S22 & S26	☐	☐	☐	☐	☐	☒
S31	Rear block electrical junction moved	S30	☐	☐	☐	☐	☐	☒
S32	Rear block cells removed	S9 & S31	☐	☐	☐	☐	☐	☒
S33	Battery support moved	S32 & S23 & S24 & S25 & S26 & S27 & S28 & S29	☐	☐	☐	☐	☐	☒

Table 3
Robot task cost duration.

Edge	*Cost	Edge	*Cost	Edge	*Cost	Edge	*Cost	Edge	*Cost
Start - S1	5.62	S2 – S1	5.645	S3 – S1	5.61	S1 - S4	0.505	S5	0.66
Start - S2	0.925	S1 – S2	0.948	S3 – S2	0.925	S2 - S4	0.515	S6	1.6
Start - S3	0.5	S1 – S3	0.502	S2 – S3	0.51	S3 – S4	0.502	S8	0.158
S9	2.352	S10	1	S11	0.997	S12	1.025	S13	1.048
S14	8.378	S20 – S14	8.537	S15	9.883	S21 – S15	10.042	S16	1
S20	0.71	S14 – S20	0.868	S15 – S20	0.868	S21	0.717	S17	0.98
S23	0.318	S24	0.362	S15 – S21	0.875	S14 – S21	0.875	S18	0.992
S25	0.367	S26	0.237	S27	0.238	S28	0.49	S19	0.997
S29	0.49	S30	16.033	S31	0.352	S32	3.862	S22	0.325
S33	0.325								

* Cost refers to the time taken to complete the task

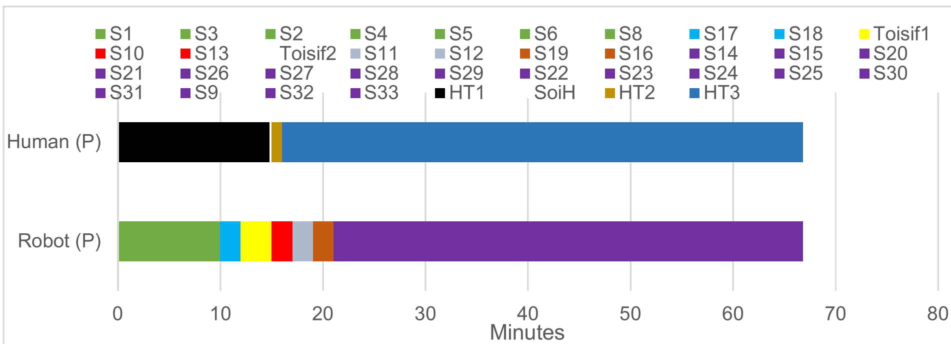

Fig. 8. Planning resulting from the proactive planner (P).

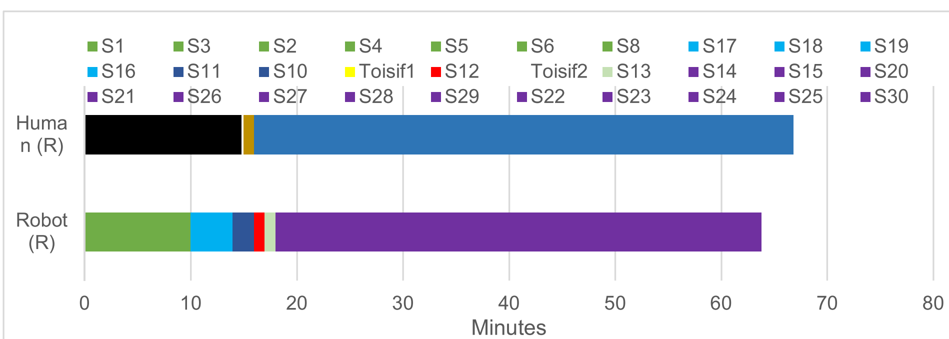

Fig. 9. Planning resulting from the reactive planner (R).

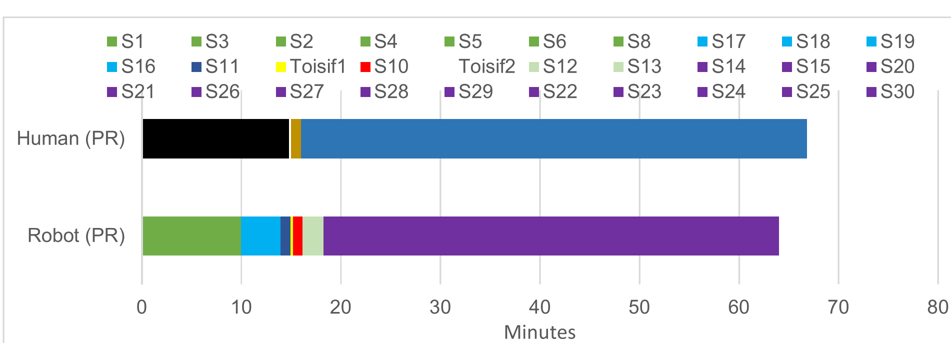

Fig. 10. Planning resulting from the proactive-reactive planner (PR).

In addition to the different tasks presented in Table 2 above, different human tasks are added to represent human non-availability denoted as HT1, which means the human partner is not available to do the prescribed work. Moreover, the human working scenario is denoted as HT2; it means the human in the workspace is doing his prescribed task and HT3 means the human partner is given another task outside the collaborative workspace. SoiH task is a virtual task that represents human idle time. It means that the human is ready to enter the collaborative workspace but has to wait for the robot to finish its tasks. Other virtual tasks defining robot idle time are also presented as ToisF1 and Toisif2, which refer to the robot idle time before and after its collaboration with human. For illustration purposes, Ts1 tasks are represented in green. The Ts1 set of tasks is defined as S1-S3-S2-S4-S5-S6-S8. Ts2 tasks as described in Section 3.1 can have either T2P task type and/or T2PN task type. T2PN task category is represented in blue (S-17-S18), as shown in Fig. 8. The T2P task category is executed as Ts2 type task represented in a given color, as shown in Fig. 9 (S11, S10). The Ts3 type task is represented in red as shown in Fig. 8 (S10, S13). Like the Ts2 task set, Ts4 task set can contain tof T2P and/or T2PN tasks. T2P tasks executed as Ts4 tasks are presented in Fig. 8 (S11, S12). T2PN tasks executed as Ts4 tasks have the same color as defined in Fig. 8 (S19, S16). For simplification, The formulation of Ts4 tasks will be related to a TSP problem formulation of the remaining T2P and T2PN tasks combined with a fixed defined set of tasks called Enf. This sequence of tasks has been determined offline using the TSP formulation applied without considering T2P or T2PN tasks. The Enf task is defined as S14-S15-S20-S21-S26-S27-S28-S29-S22-S23-S24-S25-S30-S31-S9-S32-S33 and is represented with the same color in Figs. 8, 9, 10. For comparison purpose, three parameters are considered as follows:

- ➢ The idle time of the human worker is referred to as SoiH. As they hold a significant importance in the industry; therefore, their idle time needs to be minimized.
- ➢ The total robot idle time refers to the extended time when the robot is not working. It is denoted as TsiR.
- ➢ The total production time related to one cycle activity is referred as Tpro.
- ➢ When a task is initiated in the shared workspace mode (collaborative mode) and the departure of the human is detected before the end of the task, the time spent by the robot in the collaborative mode while the human departure is detected is called as Texh because of the fact that tasks are not preemptible.

The use case presented in Fig. 8 shows that the proactive planner ends up with the highest production time of 66.477 minutes. Moreover, the human idle time is the lowest of the three as it is evaluated at 0 minutes. However, the robot idle time is the largest of all (3.08 minutes). Furthermore, Texh has the biggest (1.048 minutes). The proactive planner ensures that the human idle time is minimal by increasing the robot idle time. This affects on the total cost of production. The reactive scheduler presented in Fig. 9, has the lowest production time of about 63.449 minutes. This scheduler optimises the robot idle time, which in this case it is estimated at 0 minutes. However, the human idle time is not optimized as it is set to 0.966 minutes. The proposed proactive–reactive scheduler presented in Fig. 10 ended up with a total production time of 63.664 minutes; it is not the lowest, but there was a compromise between the human idle time of 0.181 minutes and the robot idle time of 0.215 minutes. Thus, for the purpose of optimizing the time production while minimizing human and robot idle time, the proposed scheduler has the best results.

In order to accurately analyze the real improvement of the proposed approach, 20 use cases are analyzed and presented in Table 4: Use cases results.

The results presented in Table 4 indicate that usage of reactive or proactive-reactive controller may be optimal for certain situations (cases 2, 4, 5, 7, 9, 10, 12, 15, 16, 19, and 20). In cases where proactive planning leads to suboptimal results due to overestimation predictions, the proactive-reactive (PR) planner can provide better outcomes through its reactive capabilities. So, it appears that the proactive planner is more resilient when accounting for overestimated human task time distribution. Furthermore, in certain instances (6, 11, 14) where all planners achieved the same production time, the proactive-reactive planner obtained the lowest human idle time and Texh values. Texh denotes the planner's capacity to reduce the prolonged duration of a task in shared space mode after the human collaborator leaves. In Case 17, the proposed (PR) approach achieved the best planning results in regard to production time, human, and robot idle time. In cases 1, 3, 8, 13, and 18, the most favorable trade-off between overall production time and the idle time of both human workers and robots is attained with the suggested (PR) method.

6. Limitation of the study

The study's limitation lies in the fact that the proposed planner was devised using a simulation-based method and has not been tested in an industrial case scenario. Additionally, the proposed scheduling approach has not assessed all essential cases.

7. Conclusion and future works

This paper presents a planning approach dedicated to problems where the human intervenes at precise moments with variable and non-deterministic availability and intervention times (not following a specific statistical distribution). The planning approach is based on a mix-up between a first component called offline proactive approach which takes as input the temporal predictions of the human, i.e., his availability time and his intervention duration (hypothesis 1 confirmed). The offline planning problem is reformulated as a traveling salesman type subproblem to determine among the set of tasks, the optimal sequence leading to the outcome, together with a 0/1-knapsack type problem that defines the set of tasks to be considered for the planning operation-based on task time constraints (hypothesis 2 is confirmed). This offline structure is then corrected online by a so-called reactive approach that, in the event of a deviation or discrepancy between the planned and the actual human availability and intervention time, deterministically selects the task among the planned and admissible set of tasks that presents the lowest cost in terms of execution, thereby confirming hypothesis 3.). This so-called proactive-reactive approach gives satisfactory results compared to a purely proactive approach because it is less sensitive to the quality of the prediction. Moreover, compared to a purely reactive approach, it is the one that offers the best compromise between minimizing the idle time of the human and the robot and the total production time.

However, the analysis presented in this article is based on a consideration where the so-called parallel tasks have a nominal operating time that is independent of whether the robot is in activity sharing mode or not. An extension of this work would be to explore the applicability of this approach in the case of time variations of parallel tasks that occur when switching to activity sharing mode or not. Moreover, the scheduling approach considers that the human intervenes only once in the interaction loop. For much more complex formulations with different human intervention during a cycle activity, the formulation can be evaluated with cases involving several humans and robots in the workspace. Additionally, the proposed scheme can be further improved by considering the ergonomic factor in its formulation so as to have a formal mathematical model of the change in human capabilities during

Table 4
Use cases results.

Use Case	Approach	Before collaboration	During collaboration	After collaboration	Texh	SoiH	TsiR	Tpro
Case 1 (Tdp = 12, Tip = 3, Tdr = 14.813, Tir = 1)	P	Ts1-S17-S18	S10 -S13	S11-S12-S19-S16-Enf	1.048	0	3.028	66.477
	R	Ts1-S17-S18-S19-S16-S11-S10	S12	S13-Enf	0.025	0.966	0	63.449
	PR	Ts1-S17-S18-S19-S16-S11	S10	S12-S13-Enf	0	0.181	0.215	63.664
Case 2 (Tdp = 18, Tip = 7, Tdr = 10.81, Tir = 6)	P	Ts1-S17-S18-S19-S16-S11-S10-S12	S13	Enf	0	5.991	4.742	68.401
	R	Ts1-S17-S18	S11-S10-S12-S13	S16-S19-Enf	0	0.972	1.930	65.379
	PR	Ts1- S17-S18	S13-S11-S10- S12	S16 -S19-Enf	0	0.972	1.930	65.379
Case 3 (Tdp = 10, Tip = 3, Tdr = 10.81, Tir = 2)	P	Ts1	S10-S13	S11-S12-S17-S18-S19-S16-Enf	0.048	0	1	64.449
	R	Ts1-S17-S18	S11-S10-S12	S13-S12-S19-S16-Enf	1.022	0.972	0	63.449
	PR	Ts1-S17	S10-S13	S11-S12-S18-S19-S16-Enf	0.048	0.167	0	63.636
Case 4 (Tdp = 14, Tip = 4, Tdr = 11.81, Tir = 2)	P	Ts1-S17-S18-S19-S16	S11-S10-S12	S13-Enf	1.022	1.969	0	63.449
	R	Ts1-S17-S18-S19	S11-S10-S12	S13-S16-Enf	1.022	0.969	0	63.449
	PR	Ts1-S17-S18-S19	S11-S10-S12	S13-S16-Enf	1.022	0.969	0	63.449
Case 5	P	Ts1-S17-S18-S19-S16-S11	S10-S12-S13	Enf	1.073	0.966	0	63.449
(Tdp = 15, Tip = 6, Tdr = 13.81, Tir = 2)	R	Ts1-S17-S18-S19-S16-S11	S10-S12	S13-Enf	0.025	0.966	0	63.449
	PR	Ts1-S17-S18-S19-S16-S11	S10-S12	S13-Enf	0.025	0.966	0	63.449
Case 6 (Tdp = 18, Tip = 8, Tdr = 11.81, Tir = 1)	P	Ts1-S17-S18-S19-S16-S11-S10-S12	S13	Enf	0.048	4.991	0	63.449
	R	Ts1-S17-S18-S19	S11-S10	S12-S13-Enf	0.997	0.969	0	63.449
	PR	Ts1-S17-S18-S19	S13	S10-S11-S12-S16-Enf	0.048	0.969	0	63.449
Case 7 (Tdp = 12, Tip = 5, Tdr = 10.81, Tir = 1)	P	Ts1-S17-S18	S11-S10-S12-S13	S19-S16-Enf	3.07	0.972	0	63.449
	R	Ts1-S17-S18	S11-S10	S12-S13-S19-S16-Enf	0.997	0.972	0	63.449
	PR	Ts1-S17-S18	S11-S10	S12-S13-S19-S16-Enf	0.997	0.972	0	63.449
Case 8 (Tdp = 14, Tip = 9, Tdr = 15.81, Tir = 1)	P	Ts1-S17-S18-S19-S16	S11-S10-S12-S13	Enf	3.07	0	2.031	65.480
	R	Ts1-S17-S18-S19-S16-S11-S10-S12	S13	Enf	0.048	0.991	0	63.449
	PR	Ts1-S18-S17-S19-S16-S11-S10	S12	S13-Enf	0.025	0.184	0.218	63.667
Case 9 (Tdp = 16, Tip = 8, Tdr = 11.81, Tir = 6)	P	Ts1-S18-S17-S19-S16-S11-S10	S12-S13	Enf	0	3.633	3.927	67.376
	R	Ts1-S17-S18-S19	S11-S10-S12-S13	S16-Enf	0	0.969	1.93	65.379
	PR	Ts1-S18-S17-S19	S11-S10-S12-S13	S16-Enf	0	0.969	1.93	65.379

(*continued on next page*)

Table 4 (*continued*)

Case 10 (Tdp = 14, Tip = 8, Tdr = 11.81, Tir = 6)	P	Ts1-S17-S18-S19-S16	S11-S10-S12-S13	Enf	0	1.969	1.93	65.379
	R	Ts1-S17-S18-S19	S11-S10-S12-S13	S16-Enf	0	0.969	1.93	65.379
	PR	Ts1-S17-S18-S19	S11-S10-S12-S13	S16-Enf	0	0.969	1.93	65.379
Case 11 (Tdp = 17, Tip = 5, Tdr = 10.81, Tir = 1)	P	Ts1-S17-S18-S19-S16-S11-S10-S12	S13	Enf	0.048	5.991	0	63.449
	R	Ts1-S17-S18	S11-S10	S12-S13-S19-S16-Enf	0.997	0.972	0	63.449
	PR	Ts1-S17-S18	S13	S11-S10-S19-S16-Enf	0.048	0.972	0	63.449
Case 12 (Tdp = 14, Tip = 7, Tdr = 19.91, Tir = 2)	P	Ts1-S17-S18-S19-S16	S11-S10-S12-S13	Enf	2.07	0	6.131	69.580
	R	Ts1-S17-S18-S19-S16- S11-S10-S12-S13	N/A	Enf	0	0	2.061	67.510
	PR	Ts1-S18-S17-S19-S16- S11-S10-S12-S13	N/A	Enf	0	0	2.061	67.510
Case 13 (Tdp = 10, Tip = 5, Tdr = 10.81, Tir = 1)	P	Ts1	S11-S10-S12-S13	S18-S17-S19-S16-Enf	3.07	0	1	64.449
	R	Ts1-S17-S18	S11-S10	S12-S13-S19-S16-Enf	0.997	0.972	0	63.449
	PR	Ts1-S17	S11-S10	S12-S13-S18-S19-S16-Enf	0.997	0.167	0.187	63.636
Case 14 (Tdp = 16, Tip = 8, Tdr = 10.81, Tir = 1)	P	Ts1-S17-S18-S19-S16-S11-S10	S12-S13	Enf	1.073	4.966	0	63.449
	R	Ts1-S17-S18	S11-S10	S12-S13-S19-S16-Enf	0.997	0.972	0	63.449
	PR	Ts1-S17-S18	S12	S11-S10-S13-S19-S16-Enf	0.025	0.972	0	63.449
Case 15 (Tdp = 14, Tip = 7, Tdr = 11.81, Tir = 5)	P	Ts1-S17-S18-S19-S16	S11-S10-S12-S13	Enf	0	1.969	0.93	64.379
	R	Ts1-S17-S18-S19	S11-S10-S12-S13	S16-Enf	0	0.969	0.93	64.379
	PR	Ts1-S17-S18-S19	S11-S10-S12-S13	S16-Enf	0	0.969	0.93	64.379
Case 16 (Tdp = 11, Tip = 5, Tdr = 10.81, Tir = 2)	P	Ts1-S18	S11-S10-S12-S13	Enf	2.07	0	0.008	63.457
	R	Ts1-S17-S18	S11-S10-S12	S13-S16-S19-Enf	1.02	0.972	0	63.449
	PR	Ts1-S18-S17	S11-S10-S12	S13-S16-S19-Enf	1.02	0.972	0	63.449
Case 17 (Tdp = 12, Tip = 8, Tdr = 15.81, Tir = 6.1)	P	Ts1-S17-S18	S11-S10-S12-S13	S19-S16-Enf	2.03	0	4.028	69.507
	R	Ts1-S17-S18-S19-S16- S11-S10-S12-S13	N/A	Enf	0	0.991	5.052	68.501
	PR	Ts1-S17-S18-S19-S16- S11-S10	S13-S12	Enf	0	0.181	4.027	67.691
Case 18 (Tdp = 11, Tip = 10, Tdr = 16.813, Tir = 11.1)	P	Ts1-S18	S11-S10-S12-S13	S17-S19-S16-Enf	0	0	13.04	76.487
	R	Ts1-S17-S18-S19-S16- S11-S10-S12-S13	N\A	Enf	0	0	11.1	74.549
	PR	Ts1-S17-S18-S19-S16- S11-S10-S12	S13	Enf	0	0.186	10.05	73.696
Case 19 (Tdp = 13, Tip = 8, Tdr =	P	Ts1-S18-S17-S19	S11-S10-S12-S13	S16-Enf	3.07	1.969	0	63.449
	R	Ts1-S17-S18	S11-S10	S12-S13-S19-S16-Enf	0.997	0.972	0	63.449

(*continued on next page*)

Table 4 (*continued*)

10.813, Tir = 1)	PR	Ts1-S17-S18	S11-S10	S12-S13-S19-S16-Enf	0.997	0.972	0	63.449
Case 20 (Tdp = 20, Tip = 9, Tdr = 11.813, Tir = 1)	P	Ts1-S17-S18-S19-S16- S11-S10-S12-S13	N/A	Enf	0	6.039	1	64.449
	R	Ts1-S17-S18-S19	S11-S10	S12-S13-S16-Enf	0.997	0.969	0	63.449
	PR	Ts1-S17-S18-S19	S11-S10	S12-S13-S16-Enf	0.997	0.969	0	63.449

work operations in a given time. Finally, the proposed scheme can be implemented in real industrial scenario.

Funding

This work received financial support from the Fonds de recherche du Québec—Nature et technologies (FRQNT), under grant number 2020-CO-275043 (Ramy Meziane), NSERC Discovery grant number RGPIN-2018-06329 (Martin Otis) and MITACS Acceleration IT29421. This project uses the infrastructure obtained by RoboDK, by the Ministère de l'Économie et de l'Innovation (MEI) du Quebec, John R. Evans Leaders Fund of the Canadian Foundation for Innovation (CFI) and the Infrastructure Operating Fund (FEI) under the project number 35395.

CRediT authorship contribution statement

Gilde Vanel Tchane Djogdom: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Supervision, Visualization, Writing – review & editing, Writing – original draft. **Ramy Meziane:** Funding acquisition, Project administration. **Martin J.-D. Otis:** Conceptualization, Methodology, Resources, Funding acquisition, Project administration, Supervision, Visualization.

Declaration of competing interest

There is no conflict of interest regarding the methodology and the results presented in this paper.

Data availability

No data was used for the research described in the article.

References

[1] W. Ren, et al., The decision-making framework for assembly tasks planning in human–robot collaborated manufacturing system, Int. J. Comput. Integr. Manuf. (2022) 1–19.
[2] A. Raatz, et al., Task scheduling method for HRC workplaces based on capabilities and execution time assumptions for robots, CIRP Ann. 69 (1) (2020) 13–16.
[3] P. Lou, et al., Multi-agent-based proactive–reactive scheduling for a job shop, Int. J. Adv. Manuf. Technol. 59 (2012) 311–324.
[4] I.E. Makrini, et al., Task allocation for improved ergonomics in human-robot collaborative assembly, Interact. Stud. 20 (1) (2019) 102–133.
[5] S.M. Rahman, Y. Wang, Mutual trust-based subtask allocation for human–robot collaboration in flexible lightweight assembly in manufacturing, Mechatronics 54 (2018) 94–109.
[6] T. Yu, J. Huang, Q. Chang, Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning, J. Manuf. Syst. 60 (2021) 487–499.
[7] A. Ali, et al., Heterogeneous human–robot task allocation based on artificial trust, Sci. Rep. 12 (1) (2022) 1–15.
[8] F. Fusaro, et al., A human-aware method to plan complex cooperative and autonomous tasks using behavior trees, in: IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), 2021, pp. 522–529.
[9] A. Angleraud, et al., Coordinating shared tasks in human-robot collaboration by commands, Front. Robot. AI 8 (2021) 734548.
[10] J. Kinugawa, et al., Adaptive task scheduling for an assembly task coworker robot based on incremental learning of human's motion patterns, IEEE Robot. Autom. Lett. 2 (2) (2017) 856–863.
[11] R. Alami, et al., Task planning for human-robot interaction, in: Proceedings of the 2005 Joint Conference On Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages And Technologies, 2005.
[12] G. Evangelou, et al., An approach for task and action planning in human–robot collaborative cells using AI, Proc. CIRP 97 (2021) 476–481.
[13] N. Kousi, et al., AI based combined scheduling and motion planning in flexible robotic assembly lines, Proc. CIRP 86 (2019) 74–79.
[14] N. Nikolakis, et al., Dynamic scheduling of shared human-robot manufacturing operations, Proc. CIRP 72 (2018) 9–14.
[15] H. Ding, M. Schipper, B. Matthias, Optimized task distribution for industrial assembly in mixed human-robot environments-Case study on IO module assembly, in: 2014 IEEE International Conference On Automation Science and Engineering (CASE), IEEE, 2014.
[16] A. Noormohammadi-Asl, et al., Task selection and planning in human-robot collaborative processes: to be a leader or a follower?, in: 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) IEEE, 2022.
[17] M. Askarpour, et al., Formal model of human erroneous behavior for safety analysis in collaborative robotics, Robot. Comput. Integr. Manuf. 57 (2019) 465–476.
[18] S. Alirezazadeh, L.A. Alexandre, dynamic task scheduling for human-robot collaboration, IEEE Robot. Autom. Lett. 7 (4) (2022) 8699–8704.
[19] L. Johannsmeier, S. Haddadin, A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes, IEEE Robot. Autom. Lett. 2 (1) (2016) 41–48.
[20] A. Casalino, et al., Optimal scheduling of human–robot collaborative assembly operations with time petri nets, IEEE Trans. Autom. Sci. Eng. 18 (1) (2019) 70–84.
[21] S.C. Sarin, H.D. Sherali, A. Bhootra, A precedence-constrained asymmetric traveling salesman model for disassembly optimization, IIE Trans. 38 (3) (2006) 223–237.
[22] E. Balas, The prize collecting traveling salesman problem, Networks 19 (6) (1989) 621–636.
[23] D. Navtn-Chandra, The recovery problem in product design, J. Eng. Des. 5 (1) (1994) 65–86.
[24] T.L. Urban, W.C. Chiang, An optimal piecewise-linear program for the U-line balancing problem with stochastic task times, Eur. J. Oper. Res. 168 (3) (2006) 771–782.
[25] Z. Shi, et al., Minimizing completion time for order scheduling: formulation and heuristic algorithm, IEEE Trans. Autom. Sci. Eng. 14 (4) (2015) 1558–1569.
[26] M.C. Gombolay, R.J. Wilcox, J.A. Shah, Fast scheduling of robot teams performing tasks with temporospatial constraints, IEEE Trans. Rob. 34 (1) (2018) 220–239.
[27] F. Chen, et al., Optimal subtask allocation for human and robot collaboration within hybrid assembly system, IEEE Trans. Autom. Sci. Eng. 11 (4) (2013) 1065–1075.
[28] F. Chen, et al., An assembly strategy scheduling method for human and robot coordinated cell manufacturing, Int. J. Intell. Comput. Cybern. (2011).
[29] A.M. Zanchettin, et al., Prediction of human activity patterns for human–robot collaborative assembly tasks, IEEe Trans. Industr. Inform. 15 (7) (2018) 3934–3942.
[30] A. Pupa, C. Secchi, A safety-aware architecture for task scheduling and execution for human-robot collaboration, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2021.
[31] A.M. Zanchettin, Robust scheduling and dispatching rules for high-mix collaborative manufacturing systems, Flex. Serv. Manuf. J. 34 (2) (2022) 293–316.
[32] R. Maderna, et al., Flexible scheduling and tactile communication for human–robot collaboration, Robot. Comput. Integr. Manuf. 73 (2022) 102233.
[33] C.P. Chen, AND/OR precedence constraint traveling salesman problem and its application to assembly schedule generation, in: 1990 IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings, IEEE, 1990.
[34] A.F. Alkaya, E. Duman, Application of sequence-dependent traveling salesman problem in printed circuit board assembly, IEEE Trans. Compon. Packag. Manuf. Technol. 3 (6) (2013) 1063–1076.
[35] A.V. Le, et al., Complete path planning for a tetris-inspired self-reconfigurable robot by the genetic algorithm of the traveling salesman problem, Electronics 7 (12) (2018) 344.

[36] C. Liu, et al., Goal inference improves objective and perceived performance in human-robot collaboration, arXiv Preprint (2018) arXiv:1802.01780.
[37] H. Cai, Y. Mostofi, A human-robot collaborative traveling salesman problem: robotic site inspection with human assistance, in: 2016 American Control Conference (ACC), IEEE, 2016.
[38] L. Roveda, et al., Human–robot collaboration in sensorless assembly task learning enhanced by uncertainties adaptation via Bayesian Optimization, Rob. Auton. Syst. 136 (2021) 103711.
[39] M. Bottin, G. Boschetti, G. Rosati, Optimizing cycle time of industrial robotic tasks with multiple feasible configurations at the working points, Robotics 11 (1) (2022) 16.
[40] V. Serebrenny, E. Shmalko, Multicriterial analyses of Pareto-efficiency for collaborative multi-agent systems using genetic algorithm with variations, Proc. Comput. Sci. 186 (2021) 793–801.
[41] Z. Guangbao, W. Jianhua, Multi-station and multi-robot welding path planning based on greedy interception algorithm, in: 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), IEEE, 2020.
[42] M. Dewa, Optimisation of a flexible manufacturing system through the application of greedy knapsack heuristic, in: 33rd Annual Southern African Institute of Industrial Engineering Conference, Southern African Institute for Industrial Engineering, 2022.
[43] A. Muralidharan, Y. Mostofi, Distributed beamforming using mobile robots, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2016.
[44] S. Tang, et al., Multi-station test scheduling optimization method for industrial robot servo system, J. Ambient. Intell. Humaniz. Comput. (2022) 1–17.
[45] C. Chu, J. Proth, Single machine scheduling with chain: structured precedence constraints and separation time windows, IEEE Trans. Robot. Autom. 12 (6) (1996) 835–844.
[46] F. Bendali, et al., A synchronized knapsack problem, in: 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE, 2022.
[47] H. Cai, Y. Mostofi, Human–robot collaborative site inspection under resource constraints, IEEE Trans. Robot. 35 (1) (2018) 200–215.