CONFERENCE ON PRODUCTION SYSTEMS AND LOGISTICS
CPSL 2023-2

5th Conference on Production Systems and Logistics

Adaptive Multi-Priority Rule Approach To Control Agile Disassembly Systems In Remanufacturing

Marco Wurster[1], Finn Bail[1], Sebastian Behrendt[1], Gisela Lanza[1]

[1]*wbk Institute of Production Science, Karlsruhe Institute of Technology, Germany*

Abstract

End-of-Life (EOL) products in remanufacturing are prone to a high degree of uncertainty in terms of product quantity and quality. Therefore, the industrial shift towards a circular economy emphasizes the need for agile and hybrid disassembly systems. These systems feature a dynamic material flow. Besides that, they combine the endurance of robots with the dexterity of human operators for an effective and economically reasonable EOL-product treatment. Moreover, being reconfigurable, agile disassembly systems allow an alignment of their functional and quantitative capacity to volatile production programs. However, changes in both the system configuration and the production program to be processed call for adaptive approaches to production control. This paper proposes a multi-priority rule heuristic combined with an optimization tool for adaptive re-parameterization. First, domain-specific priority rules are introduced and incorporated into a weighted priority function for disassembly task allocation. Besides that, a novel metaheuristic parameter optimizer is devised to facilitate the adaption of weights in response to evolving requirements in a reasonable timeframe. Different metaheuristics such as simulated annealing or particle swarm optimization are incorporated as black-box optimizers. Subsequently, the performance of these metaheuristics is meticulously evaluated across six distinct test cases, employing discrete event simulation for evaluation, with a primary focus on measuring both speed and solution quality. To gauge the efficacy of the approach, a robust set of weights is employed as a benchmark. Encouragingly, the results of the experimentation reveal that the metaheuristics exhibit a notable proficiency in rapidly identifying high-quality solutions. The results are promising in that the metaheuristics can quickly find reasonable solutions, thus illustrating the compelling potential in enhancing the efficiency of agile disassembly systems.

Keywords

Adaptive Production Control; Metaheuristics; Disassembly; Agile System; Priority Rules

1. Introduction

Remanufacturing is a vital method to close the loop in circular supply chains by increasing material efficiency and optimizing resource consumption [1]. In this process, End-Of-Life (EOL) products undergo disassembly into individual components, which are subsequently recovered and reassembled to create a product with restored quality and functionality, resembling a "like-new" state [2]. However, the implementation of efficient remanufacturing systems comes with many domain-specific challenges that make the planning and operation of such systems more complex than in most conventional production systems [3] [4]. Especially the disassembly processes are of complex matter [5], as there is a high degree of uncertainty regarding the type, quantity and quality of incoming EOL products [6]. Therefore, disassembly processes are merely conducted manually in remanufacturing, which limits its economic feasibility, especially in high-wage countries [7].

DOI: https://doi.org/10.15488/15278
ISSN: 2701-6277

The distinctive features of product disassembly impose specific requirements on the design of efficient disassembly systems. Especially the synergetic integration of manual and automated workstations is of essence for an economical disassembly [7]. High flexibility in all components of the system is required including a modular structure, hybrid workstations, flexible material flow and flexible tools [7]. Agile disassembly systems realize these attributes by focussing on the integration of flexibly automated resources in a hybrid factory to deal with varying product specifications [8]. The concept is based on the idea of combining learning robots with human operators to increase productivity and reduce operative costs while maintaining enough flexibility to deal with inherent uncertainties. Robots with cognitive abilities and problem-solving competencies take over suitable disassembly tasks from human operators but are backed up manually in case of operational failures. A modular system structure with loosely linked disassembly stations allows for a flexible material flow to realize product instance-specific routings. Besides that, the system can adjust its capacity to volatile production programs by adding, removing or substituting stations. This enables the system to reconfigure and adapt to changing events throughout multiple production periods [8]. While the hybrid system architecture poses a prerequisite for efficient disassembly, a suitable planning and control system is vital to exploit the additional degrees of freedom.

To overcome the challenges in agile disassembly systems, a control system must aim to resolve the following requirements: The control system must (A) manage a highly flexible and dynamic material flow, as the redundancies in the system allow many routing alternatives while coping with complex system and order states. Besides that, the control system must (B) adapt to different system configurations and loads to ensure good allocation decisions in a continuously changing system environment due to frequent system reconfigurations. As the high degree of uncertainty in remanufacturing leads to ineffective production plans [9], this paper neglects predictive schedules and focuses on reactive order allocation.

2. Related Work

Disassembly planning and control is a broad research field with a rising interest in the last decades [10]. Nevertheless, there is limited research available on appropriate methods that address the specific challenges in agile disassembly systems. Available approaches are either too rigid for reactive control tasks, neglect hybrid systems or do not focus the organizational level and don't scale for disassembly systems with multiple stations. Tang et al. propose a promising solution to simultaneously control the disassembly sequence and the allocation of operations and stations [11]. The approach is dynamic in a way that it doesn't rely on an initial disassembly plan. However, tasks are not distributed among different stations, as all stations have the functional capacity to fully disassemble a discarded product. Concerning the allocation of operations to specific disassembly stations, Kim et al. state as well that predefined disassembly plans are rarely effective as actual system states mostly diverge from the planned system states [12,13]. Therefore, they propose an approach where the initial disassembly plan is rerouted in case of occupied stations or machine failures. In contrast, Hrdina and Zülch state that merely rescheduling a predefined disassembly schedule isn't sufficient to cope with the high degree of uncertainty in a disassembly system, as often systematic changes have to be implemented for further scheduling. Instead, decisions should be made dynamically and individually for each operation [14]. They introduce a dynamic control system for a manual disassembly line which enables a simulation-based optimization of operations. Stations can adjust disassembly operations or methods and operations can be shifted to subsequent stations. Paschko et al. identify that static measures for material release control such as ConWIP result in efficiency losses when applied in highly uncertain environments of disassembly systems [15]. Instead, they propose an adaptive control logic based-on reinforcement learning, which takes system information into account. In [16] a dynamic control logic is proposed for agile hybrid disassembly systems. It balances the allocation of disassembly tasks between a flexible robot and a human operator considering different quality conditions of discarded products. The approach is based on Deep Q-Learning which shows promising results in reducing operational failures and reducing operational costs and system makespan when compared with a priority rule heuristic. The investigated system and the control problem are close to the setting at hand. However, the deployed reinforcement learning agent requires retraining and tuning after the disassembly system is reconfigured, limiting its adaptivity and applicability in practice.

Heuristic priority rules (PR) are often used as a simple means to control the material flow in conventional production systems [17]. Using priority rules to make individual, self-organized and robust routing decisions (see [18]) can be suitable for agile disassembly systems. However, no singular priority rule optimizes the material flow over all possible system configurations [17]. Hence, multiple rules can be combined through a parametrization to depict a more diverse control mechanism and allow adjustments of the control via reparameterization. Typically, priority rules are applied to sequencing problems, with only a few relevant approaches explicitly exploring them for order-to-station allocation problems in flexible job shops. In [19] and [20] stations are enabled to select the next order they process from a set of pending orders. The selection is based on priority rules. Stecke et al. conclude that the effectiveness of priority rules is highly dependent on the system configuration while Xanthoplous et al. emphasize that a combination of rules yields favourable results [19,20]. In [21] and [18] an operation selects the next station it is being processed on. In these cases, the usage of priority rules leads to an increased system performance compared to a random station allocation. In the scope of product disassembly, Guide et al. present an approach, that uses priority rules for disassembly sequence planning to determine the processing order in the input buffer of a disassembly station [22]. In general, however, the usage of priority rules is very limited.

Even though disassembly control is a vibrant research topic, to the best of the authors' knowledge, only a few existing approaches are suitable to control agile disassembly systems. Priority rules on the other hand are a well-established means in flexible job shops, but, despite their promising potential, they haven't been applied to flexible disassembly shops yet. Following this research deficit, a novel adaptive control logic for agile disassembly systems based on heuristic priority rules is proposed in this work. First, domain-specific priority rules are presented, which are fused in a weighted sum to result in a joint priority score. Weights are optimized according to the target criteria, urging for near-optimal foresighted order allocation decisions in a complex disassembly environment (A). To account for adaptivity, the logic is furthermore extended by a metaheuristic optimization module to re-parameterize the control system after changes in system configuration (B).

The remainder of this paper is structured as follows: Section 3 presents the agile control approach, comprising the system architecture, the rule-based control module and the module for system re-parameterization with five different optimization algorithms. In section 4, the different algorithms are validated and compared by efficiency and solution quality. Eventually, section 5 concludes the paper with a summary and outlook on future work.

3. Approach

The proposed approach builds on the system architecture depicted in Figure 1. It encompasses a model of the agile disassembly system including an executable discrete-event simulation. Combined with the control system module determining the material flow, the *Operation* perspective is marked. Additionally, to accommodate reconfiguration capabilities, a *Reconfiguration* perspective is introduced, featuring a system configurator and a control optimizer for the control system. Unlike the control system, both modules strategically adjust the system at designated time points without direct interference during system operations. Given a production program comprising multiple subsequent production periods, the system is typically adapted between these periods in two steps. First, a new system configuration is generated by a system configurator with functionalities like capacity and layout planning, enabling structural adaption. Second, the control optimizer adapts the control system for a logical adaption using an image of the new system configuration. In the following, the focus is on the system model, the control system and the control optimizer, which will be described in detail. Due to limited scope, the system configurator will not be extensively covered in this paper. However, a compatible approach can be found in [23].

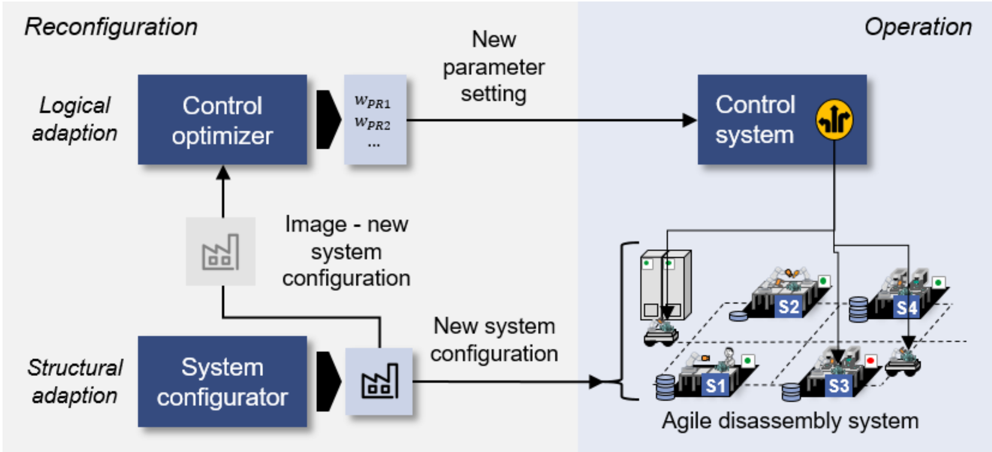

Figure 1: Overview of the system architecture

3.1 System Model

In its basic features, the agile disassembly system is based on the model established in [16]. However, the model is slightly extended, hence, modifications will be specifically highlighted in the following.

The agile disassembly system pursues to disassemble a set of orders $O = (o_1, o_2, \dots)$, predefined by a given production program and released to the system. The overall aim of the disassembly system is to maximize the number of fully disassembled orders given a defined amount of resources. To fully disassemble an order, product-specific disassembly tasks must be processed. The set $Tsks_i = (tsk_{i,1}, tsk_{i,2}, \dots)$ denotes the sum of all possible disassembly tasks throughout the entire disassembly process of order o_i. However, it is not always necessary to process all tasks in $Tsks_i$, as alternative disassembly sequences are possible. Let $Tsks_{i,possible} = (tsk_{i,1}, tsk_{i,2}, \dots)$ be the set of all disassembly tasks of order o_i that can be processed at the current time. The orders are processed by a set of stations $S = (s_1, s_2, \dots)$ which can be adjusted between production periods. Each station has a different set of capabilities so that the disassembly system can adjust its functional and operational capacities to new requirements by reconfiguring itself. The capabilities of station s_j can be denoted as $Cap_j = (tsk_1, tsk_2, \dots)$. Meanwhile, the set of stations that are capable of processing task tsk_i is noted as $Enab_i = (s_1, s_2, \dots)$. A specific instance of a task, when executed on a station, marks an operation op_i.

The simulation includes order-specific processing times and failure rates. The product structures and corresponding disassembly tasks are modelled and organized by disassembly Petri nets, similar as proposed in [11,24]. As the state of EOL products can strongly differ when entering a disassembly system, quality classes are introduced to better depict reality. The quality class influences the processing times, operation failure rates and the capability of a station to perform a task.

To account for the hybrid nature of the disassembly system, three distinct station types with varying capabilities and attributes are introduced. The first type are Manual Stations (MS), which rely on human operators and possess the ability to perform all disassembly operations. Although they offer the highest level of flexibility, operating these stations can be costly. The second type, Automatic Stations (AS), stand out due to their rapid and nearly deterministic operation times. However, their capabilities are limited to routine tasks and disassembling products that are in good condition. Conversely, Robotic (Learning) Stations (RS) gradually extend the scope of automatic resources. Equipped with flexible tools and cognitive abilities, RS are capable of assuming a broader variety of tasks. Moreover, they can effectively solve minor problems and autonomously deal with anomalies which is prerequisite for many disassembly tasks. In addition to the individual station instances, an integral part of the system is the employment of automated guided vehicles (AGV), which facilitate the transportation of orders between the stations [25].

Overall, the introduction of these three station types, namely MS, AS and RS, along with the use of AGV, enables an effective and efficient operation of the disassembly system.

3.2 Control Logic

Released disassembly orders are initially vacant and require an allocation decision including both the next disassembly task and the next station for the next disassembly operation. In most cases, multiple disassembly tasks $\forall tsk_i \in Tsks_{i,possible}$ can be processed by several stations $\forall s_j \in Enab_i$, making the decision very complex. Each allocation option can be described as a tuple (tsk_i, s_j) corresponding to a specific operation denoted $Op_{i,j}$. To select the optimal operation, multiple priority rules Π are used to calculate individual rule-specific scores $v_k \in \mathbb{R}, v_k \in [0,1]$. Thereby, each $\pi \in \Pi = [\pi_1, \pi_2, \dots, \pi_K]$ is given a weight w_k for scaling and to foster the rule-specific importance, while $\sum w_k = 1$ needs to be respected for a convex combination of the individual weights. Eventually, for each operation, a total score $v_{\text{Total}}(Op_{i,j})$ can be facilitated by a weighted sum to identify the most suitable operation:

$$v_{Total}(Op_{i,j}) = \sum_{k=1}^{K} w_k v_k \tag{1}$$

The main goal of the agile disassembly system is to unburden human operators and effectively integrate automated resources while increasing the productivity of the system. Therefore, the main performance indicators for the agile disassembly system are throughput - if time is limited - or makespan - if the order backlog is limited. Appropriate priority criteria need to balance operational and tactical preferences to improve said performance indicators. Thus, the following three rules are presented:

Lowest Buffer Utilization (LBU) scores stations by the number of orders that are in the input- and output buffer of a station. This is done by calculating the relative proportion of buffer spaces occupied. Hereby cap_j denotes the maximal buffer capacity of station s_j and $ocup_{j,in/out}$ the occupied input and output buffer.

$$v_{LBU}(Op_{i,j}) = \frac{cap_j - (ocup_{j,in} + ocup_{j,out})}{cap_j} \tag{2}$$

Lowest Station Cost (LSC) prioritises stations based on their costs. These can be hourly costs or simply relative cost rates. The station costs of station s_j are denoted as c_j. For scaling, it is set into relation with the highest station costs c^{max} and the lowest station cost stations c^{min} of all deployed stations (min-max normalization).

$$v_{SCR}(Op_{i,j}) = \frac{c^{max} - c_j}{c^{max} - c_{min}} \tag{3}$$

Finally, *Shortest Processing Time (SPT)* calculates the mean processing times of certain disassembly tasks on specific stations. Stations with shorter processing times are preferred. Unlike the previous two, SPT is a common priority rule, well-known in the scheduling literature [26]. However, while conventional approaches usually consider deterministic processing times, this approach builds on historic data to account for uncertainty. Thus, a rolling window of the last operations is used to calculate $t_{i,j}^{avg}$, the mean and expected processing time of tsk_i processed on stations of the same type as s_j. For a normalized score v_{SPT}, $t_{i,j}^{mean}$ is compared with the maximum $t_{i,\max}^{mean}$ and minimum $t_{i,\min}^{mean}$ mean processing times for tsk_i of all capable stations:

$$v_{SPT}(Op_{i,j}) = \frac{t_{i,j}^{mean}}{t_{i,max}^{mean} - t_{i,min}^{mean}} \tag{4}$$

The given priority rules have broad applicability, suitable not only for remanufacturing but also for agile production systems. However, the control system is not limited to them but can also incorporate other rules that are common in linear production. The given ones are chosen as examples due to their ability to balance system utilization and cost-effectiveness-to-throughput ratio under uncertainty, an emerging challenge closely associated with the proliferation of hybrid disassembly systems.

Through changes in the system configuration and the production program, a shift in the optimal parametrization is observable. Figures 1-3 depict the changes in the system efficiency of individual parametrizations that result from changes in the production system. All figures show the exhaustive convex solution space for all possible weight combinations, based on an individual system configuration. Each point in a figure represents a parametrization w which was simulated 20 times to account for stochastic behaviour. The colour indicates the number of orders which were disassembled during an eight-hour shift. Additionally, the red circle marks parametrizations that yield results within 2.5% of the found optima. Figure 2a corresponds to systems with a high proportion of MS in the system configuration. Figure 2b is generated from a system with many orders from low-quality classes which must be disassembled. Figure 2c depicts the influence of the order release. A ConWIP logic is used in all cases, while in case of Figure 2c, the fixed WIP-limit is reduced by half.

This paper aims to enable the control system to adapt its parametrization and thereby optimize the material flow individually for every time period. This way the flexibility of the agile disassembly system can be leveraged more optimally.

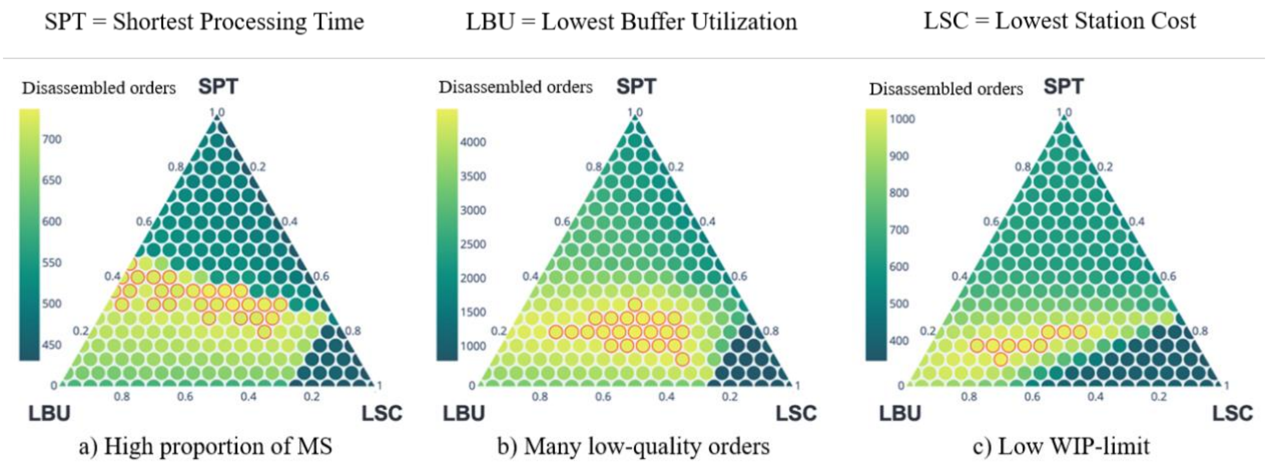

Figure 2: Amount of fully disassembled orders for different system configurations (a) High proportion of MS, b) Many low-quality orders, c) Low WIP-limit) depending on parametrization – Red border indicates parametrizations that are within 2.5% of the best solution

To re-optimize the parametric control, four different metaheuristics are implemented and compared against a robust parameter set. The metaheuristics can be classified as population-based and local search methods [27]. A grid search was applied to all metaheuristics to determine the optimal parameter configuration.

Simulated Annealing (SA) is a local Search Metaheuristic that is based on the cooling behaviour of metals [28]. It distinguishes itself from other local search methods by accepting worse solutions with a defined probability influenced by an iteratively changing temperature. By gradually decreasing the temperature, the algorithm transitions from exploration to exploitation. Contrarily Particle Swarm Optimization (PSO), Evolutionary Algorithms (EA) and Artificial Bee Colony (ABC) are population-based metaheuristics, however, all of them utilize different search strategies. PSO mimics the movement of animal groups by moving the population throughout the search space [29]. Hereby the movement is influenced by the best solution for each entity and the best solution for the population. EA is based on the principle of natural selection [30]. Through three basic operations; selection, crossover and mutation, the fitness of the population is incrementally increased. Lastly, ABC additionally combines aspects from local search and random search as the neighbourhoods of the best solutions are predominantly searched and non-improving solutions are periodically reinitialized randomly [31].

4. Validation

To evaluate the effectiveness of a dynamic reparameterization of the production control, six use cases were defined each simulating an eight-hour shift. The use cases differ in their production program, machine setup and release mechanism. Due to the high complexity of fully depicting all modifications and the limited scope of the paper, only a qualitative description of the modifications is given in Table A1. All simulations consider six different products for disassembly, each with a different number of quality classes (ranging between 3 and 5), which additionally influence processing times and the capabilities of machines. As mentioned above the costs to operate MS (10 cost units) are costlier than RS (7 cost units) and AS (5 cost units). The products are also distinguishable through different degrees of complexity as the number of components and alternative disassembly sequences can change. To benchmark the effectiveness of the optimizers, the results are compared with a robust, however static parametrization. This parametrization corresponds to $\mathrm{w} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ and was determined by selecting the best static parametrization throughout the six use cases. The hyperparameters for each metaheuristic were optimized through a grid search in an area guided by related literature and preliminary experiments. Each optimizer is evaluated through the mean number of fully disassembled orders which is calculated by a sample of five simulation replications per use case with varying seeds to account for stochasticity. The overall increase in system efficiency is presented in Table 1.

Table 1: Mean number of orders fully disassembled by metaheuristic after optimization – Percentual efficiency increases compared to robust parameters indicated in brackets

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Mean
Robust Parameters	732.2	1016.8	1474.4	578.6	4243.2	741.4	**0.00%**
Simulated Annealing	718.4 (-1.88%)	1053.8 (3.64%)	1550.2 (5.14%)	670.2 (15.83%)	4490.0 (5.82%)	1029.6 (38.87%)	**11.24%**
Particle Swarm Optimization	739.8 (1.04%)	1063.6 (4.60%)	1512.4 (2.58%)	660.6 (14.17%)	4523.4 (6.60%)	1013.4 (36.69%)	**10.95%**
Artificial Bee Colony Optimization	729.2 (-0.41%)	1087.0 (6.90%)	1550.2 (5.14%)	664.8 (14.90%)	4514.4 (6.39%)	1031.6 (39.14%)	**12.01%**
Evolutionary Algorithm	705.4 (-3.66%)	998.8 (-1.77%)	1543.6 (4.69%)	624.4 (7.92%)	4365.2 (2.88%)	984.6 (32.80%)	**7.14%**

Firstly, all optimizers outperform static parametrization, which indicates that adaptive control is beneficial. The degree of improvement strongly depends on the considered use case. This can be explained by the fact that in some cases the robust parameters are close to the optimum and further improvement is simply not possible (e.g., use case 1). However, in other cases a change in parametrization results in large improvements (e.g., use case 6). Especially in cases where the RS and AS can only execute a small proportion of the overall tasks, it is vital to utilize these stations as much as possible. This often requires case specific adjustments to the parametrization, hence creating possibilities of improvement through optimization. Focussing on the mean improvement, the ABC provides the best results and the EA the smallest amount of improvement. Nevertheless, except for the EA, all optimizers are within a span of 2%. This motivates to additionally take the convergence speed of each algorithm into account.

Figure 3 exemplary illustrates the development of the current best-found solution throughout the algorithm. The fitness value is plotted against the number of performed simulation runs. As each evaluation of a parametrization requires one simulation run, this unit of measurement allows comparing the different metaheuristics. An analysis based on the number of iterations isn't possible since population-based approaches require multiple function evaluations per iteration compared to local search methods.

The fitness values of Figure 3 do not strictly mandate the results that are seen in Table 1. As the simulation is stochastic, a high fitness during optimization does not inevitably ensure a good outcome in the actual production period. The convergence speed is also quantitively analysed in Table A2 by taking the mean number of simulation runs required to overcome the threshold of within 2% of the best-found fitness value. It can be said that the EA yields the slowest improvement of the optimizers, which confirms the prior findings from Table 1. The results of SA, ABC and PSO are comparable. SA seems to converge faster than the population-based approaches throughout the use cases 1-5, but the discrepancy in use case 6 is significant. This could be interpreted as a statistical outlier or a possible edge case described in use case 6. Nevertheless, SA and PSO demonstrate a high convergence speed on average.

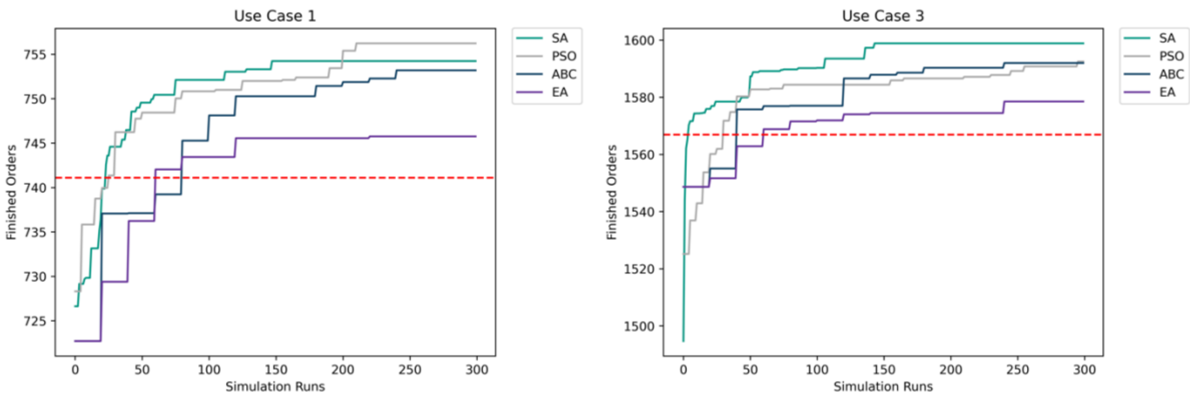

Figure 3: Mean fitness value of the best-found solution - Red line indicates the threshold within 2% of the global best-found solution

5. Conclusion

This paper comprises a model and control system for agile hybrid disassembly systems in remanufacturing. In order to boost the system efficiency, a combined priority rule approach that reactively allocates disassembly orders to suitable stations has been proposed. The approach shows promising results while posing a suitable method for industrial practitioners due to its intelligible nature and domain-specific expandability. Besides that, the combination of multiple priority rules is to be highlighted at this point, as it yields better results than conventional uni-criterial approaches in complex environments such as the disassembly domain. Besides that, it is shown that logical adaptions must follow structural adaptions in case of a system reconfiguration, for which reason, the approach was extended by a simulation-based metaheuristic parameter optimizer. While all reparameterizations outperform the robust parameters, ABC produces the best results. However, PSO and SA require less time to achieve a reasonable outcome.

While preliminary experiments indicated that the selected priority rules (LSC, LBU, SPT) yield good results, further rules should be incorporated to potentially enhance effectiveness. In particular, a multi-priority rule heuristic based on more than three rules should be investigated. Besides that, a comparison with more sophisticated generalized control approaches such as RL-based allocation agents could reveal interesting findings. Eventually, the approach should be validated on a real scenario to bridge the gap to industrialization.

Acknowledgements

This work is part of the AgiProbot project funded by the Carl Zeiss Foundation.

Appendix

Additional Tables

Table A1: Overview of the modifications that distinguish the considered use cases

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Modification	Increased frequency of products in bad condition	Increased frequency of products with high complexity	Increased frequency of products with low complexity	Only products in bad condition	High proportion of MS in system configuration	Lower CONWIP level

Table A2: Mean number of simulation runs required to overcome the threshold of 2% within the best found solution for each metaheuristic

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Mean
Simulated Annealing	23	44	4	17	3	201	**48.67**
Particle Swarm Optimization	25	50	30	25	15	85	**38.33**
Artificial Bee Colony	80	140	40	80	20	140	**83.33**
Evolutionary Algorithm	60	320	60	180	60	100	**130**

References

[1] Potting, J., Hekkert, M., Worrel, E., Hanemaaijer, A., 2017. Circular Economy: measuing innovation in the product chain. Planbureau voor de leefomgevig (2544).

[2] Giutini, R., Gaudette, K., 2003. Remanufacturing: The next great opportunity for boosting US productivity. Business Horizons 46 (6), 41–48.

[3] Zhang, R., Ong, S.K., Nee, A.Y.C., 2015. A simulation-based genetic algorithm approach for remanufacturing process planning and scheduling. Applied Soft Computing 37 (6), 521–532.

[4] Cui, Y., Guan, Z., He, C., Yue, L., 2017. Research on remanufacturing scheduling problem based on critical chain management. IOP Conf. Ser.: Mater. Sci. Eng. 215, 12005.

[5] Guide, V.D.R., 2000. Production planning and control for remanufacturing: industry practice and research needs. Journal of Operations Management 18 (4), 467–483.

[6] Lundmark, P., Sundin, E., Björmark, M., 2009. Industrial Challenges within the Remanufacturing System. Proceedings of The International 3rd Swedish Production Symposium, SPS'09, Göteborg, Sweden, 2-3 December 2009, 132–138.

[7] Kim, H.-J., Harms, R., Seliger, G., 2007. Automatic Control Sequence Generation for a Hybrid Disassembly System. IEEE Trans. Automat. Sci. Eng. 4 (2), 194–205.

[8] Lanza, G., Asfour, T., Beyerer, J., Deml, B., Fleischer, J., Heizmann, M., Furmans, K., Hofmann, C., Cebulla, A., Dreher, C., Kaiser, J.-P., Klein, J.-F., Leven, F., Mangold, S., Mitschke, N., Stricker, N., Pfrommer, J., Wu, C., Wurster, M., Zaremski, M., 2022. Agiles Produktionssystem mittels lernender Roboter bei ungewissen Produktzuständen am Beispiel der Anlasser-Demontage. at - Automatisierungstechnik 70 (6), 504–516.

[9] Kurilova-Palisaitiene, J., Sundin, E., 2014. Challenges and Opportunities of Lean Remanufacturing. IJAT 8 (5), 644–652.

[10] Priyono, A., Ijomah, W., Bititci, U., 2016. Disassembly for remanufacturing: A systematic literature review, new model development and future research needs. JIEM 9 (4), 899.

[11] Tang, Y., Zhou, M., Caudill, R.J., 2001. An integrated approach to disassembly planning and demanufacturing operation. IEEE Trans. Robot. Automat. 17 (6), 773–784.

[12] Kim, H.-J., Chiotellis, S., Seliger, G., 2009. Dynamic process planning control of hybrid disassembly systems. Int J Adv Manuf Technol 40 (9-10), 1016–1023.

[13] Kim, H.-J., Ciupek, M., Buchholz, A., Seliger, G., 2006. Adaptive disassembly sequence control by using product and system information. Robotics and Computer-Integrated Manufacturing 22 (3), 267–278.

[14] Hrdina, J., Zülch, G., 2017. Simulationsunterstützende Entwicklung von Methoden zur reaktiven Steuerung von Demontagelinien. Simulation in produktion und Logistik, 325–334.

[15] Paschko, F., Knorn, S., Krini, A., Kemke, M., 2023. Material flow control in Remanufacturing Systems with random failures and variable processing times. Jnl Remanufactur 13 (2), 161–185.

[16] Wurster, M., Michel, M., May, M.C., Kuhnle, A., Stricker, N., Lanza, G., 2022. Modelling and condition-based control of a flexible and hybrid disassembly system with manual and autonomous workstations using reinforcement learning. J Intell Manuf 33 (2), 575–591.

[17] Geiger, C.D., Uzsoy, R., Aytuğ, H., 2006. Rapid Modeling and Discovery of Priority Dispatching Rules: An Autonomous Learning Approach. J Sched 9 (1), 7–34.

[18] Scholz-Reiter, B., Peters, K., Beer, C. de, 2004. Autonomous control of shop floor logistics. Proc. IFAC-MIM, in print.

[19] Stecke, K.E., Solberg, J.J., 1981. Loading and control policies for a flexible manufacturing system. International Journal of Production Research 19 (5), 481–490.

[20] Xanthopoulos, A.S., Koulouriotis, D.E., Gasteratos, A., Ioannidis, S., 2016. Efficient priority rules for dynamic sequencing with sequence-dependent setups. 10.5267/j.ijiec, 367–384.

[21] Subramaniam, V., Lee, G.K., Ramesh, T., Hong, G.S., Wong, Y.S., 2000. Machine Selection Rules in a Dynamic Job Shop. Int J Adv Manuf Technol 16 (12), 902–908.

[22] Guide, V.D.R., Kraus, M.E., Srivastava, R., 1997. Scheduling policies for remanufacturing. International Journal of Production Economics 48 (2), 187–204.

[23] Behrendt, S., Wurster, M., May, M.C., Lanza, G., 2023. Extended Production Planning of Reconfigurable Manufacturing Systems by Means of Simulation-based Optimization. Hannover : publish-Ing.

[24] Zussman, E., Zhou, M., 1999. A methodology for modeling and adaptive planning of disassembly processes. IEEE Transactions on Robotics and Automation 15 (1), 190–194.

[25] Klein, J.-F., Wurster, M., Stricker, N., Lanza, G., Furmans, K., 2021 - 2021. Towards Ontology-based Autonomous Intralogistics for Agile Remanufacturing Production Systems, in: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). 2021 IEEE 26th International Conference on Emerging Technologies and Factory Automation (ETFA), Vasteras, Sweden. 07.09.2021 - 10.09.2021. IEEE, pp. 1–7.

[26] Haupt, R., 1989. A survey of priority rule-based scheduling. OR Spectrum 11 (1), 3–16.

[27] Sörensen, K., Glover, F.W., 2013. Metaheuristics, in: Fu, M.C., Gass, S.I. (Eds.), Encyclopedia of Operations Research and Management Science. Springer, Boston, MA, pp. 960–970.

[28] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science (New York, N.Y.) 220 (4598), 671–680.

[29] Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks. ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia. 27 Nov.-1 Dec. 1995. IEEE, pp. 1942–1948.

[30] Jones, G., 2002. Genetic and Evolutionary Algorithms, in: Ragu Schleyer, P. von, Allinger, N.L., Clark, T., Gasteiger, J., Kollman, P.A., Schaefer, H.F., Schreiner, P.R. (Eds.), Encyclopedia of Computational Chemistry, vol. 1. John Wiley & Sons, Ltd, Chichester, UK, p. 1.

[31] Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization TR06. Erciyes university engineering Faculty, Kayseri, Turkey.

Biography

Marco Wurster (*1993) is a research associate at the Institute of Production Science (wbk) at the Karlsruhe Institute of Technology (KIT) working in the group of production system planning. In his research he focuses on production planning and control of agile production systems in environments with a high degree of uncertainty such as remanufacturing and product disassembly.

Finn Bail (*1998) is a research associate at the Institute of Production Science (wbk) at the Karlsruhe Institute of Technology (KIT) working in the group of production system planning. He studied industrial engineering and management at the Karlsruhe Institute of Technology (KIT).

Sebastian Behrendt (*1997) is a research associate at the Institute of Production Science (wbk) at the Karlsruhe Institute of Technology (KIT) working in the group of production system planning. He studied mechanical engineering with a specialization in production systems, data analysis and machine learning at the Karlsruhe Institute of Technology (KIT).

Gisela Lanza (*1973) is member of the management board at the Institute of Production Science (wbk) of the Karlsruhe Institute of Technology (KIT). She heads the Production Systems division dealing with the topics of global production strategies, production system planning, and quality assurance in research and industrial practice. Her research focus is on the holistic design and evaluation of production systems.