	Procedure	Source	Challenges							
			Stochastic processroutes because of fluctuating core quality	*Explanation*	*arbitrary failure rates of cores*	*Explanation*	*Variable processing times*	*Explanation*	*technical incompatibilities causing variable transition times*	*Explanation*
Heuristic	Forward Scheduling	Wiendahl (2019) 978-3-446-44661-8	**1**	Forward scheduling requires a fixed work plan (fixed number of operations) in advance. If it only becomes clear during processing (e.g. after step 2) that an additional step ('welding') is necessary, the originally calculated end date becomes invalid. The method has no mechanism for 'conditional paths'. Xu et al. (2024) describe this as the main reason why static scheduling models in remanufacturing must be replaced by 'integrated process planning'. https://doi.org/10.3233/JIFS-233408	**1**	Forward scheduling calculates the order linearly. If a core fails in step 3 (reject), the run ends there in reality. The process itself does not 'notice' this and schedules fictitious capacities for the subsequent steps until the order is cancelled manually.	**2**	Rigid buffering is necessary. Forward scheduling uses fixed planned times from the master data. In order to cushion the variance (e.g. rust removal takes 10–50 minutes), the master data must be inflated with extremely high safety margins (buffers). If you plan using worst-case times, the delivery date is artificially pushed far into the future (high planned lead time), even though the order could often be completed much faster. This leads to unnecessarily high inventories. --> According to Lödding, this also leads to poor adherence to delivery dates.	**1**	Ignoring blockages. If a part is held up due to a lack of spare parts or clarification requirements, the transition time explodes. Forward scheduling only adds the planned transition time.
	Backward Scheduling	Wiendahl (2019) 978-3-446-44661-8	**1**	Backward scheduling is based on a fixed work plan. If an additional step ('welding') is required in remanufacturing with a probability of 30%, the deterministic model ignores this time requirement. The calculated start date is too late. If the additional step occurs, the delivery date can no longer be met because there is no buffer ('slack').	**1**	Backward scheduling plans for one order. If the core fails in the middle of the process (at step 3), the order is 'dead'. A new order would have to be started immediately. Since backward scheduling plans 'on the edge' (latest start), there is no time left to bring in a replacement core. The delivery date is immediately missed. Li et al. (2015) argue that random yields necessarily require buffer stocks or earlier start times, which contradicts the 'just-in-time' logic of backward scheduling. https://doi.org/10.1016/j.cor.2014.01.005	**2**	To be on the safe side, companies often use 'safety lead times' (buffers) for backward scheduling. In remanufacturing, however, the variance (standard deviation) is so high that the buffers would have to be enormous. The calculated start date often slips into the past ('We should have started last week'). The system generates unrealistic 'rush orders' even before the work begins.	**1**	Backward scheduling assumes 'smooth' transitions. If a part is awaiting clarification (incompatibility), the process comes to a standstill.
	Midpoint Scheduling	Wiendahl (2019) 978-3-446-44661-8	**1**	In remanufacturing, it is not clear which work step will become the bottleneck. Cleaning may be the bottleneck for 70% of cores, but for 30% of poor-quality cores, corrosion protection coating is the bottleneck. The 'fixed' bottleneck is incorrect. The plan is disrupted by unforeseen bottlenecks (at other stations). A classic example from the literature: Diagnosis is set as the bottleneck, but 10% of the cores require a completely different route instead of diagnosis → replacement. Xue et al. (2021) show that bottlenecks in remanufacturing shift dynamically and that static fixation leads to suboptimal plans. 10.1007/s11356-021-15233-2	**1**	Centralised scheduling attempts to utilise the bottleneck to 100% capacity ('constraint exploitation'). However, if 30% of the cores fail and leave the bottleneck earlier, capacity suddenly becomes available that is not being used. The system cannot respond flexibly. The upstream machines are inhibited (because they are waiting for the bottleneck), while the downstream machines are starved (because fewer finished parts are coming in).	**1**	If an upstream operation takes longer than planned (rusty part = extra grinding), arrival at the bottleneck is delayed. This results in underutilisation of the bottleneck. The plan, which was based on bottleneck utilisation, becomes invalid. A new bottleneck arises upstream. The entire schedule is obsolete.	**1**	If a part remains pending clarification (e.g. question: 'Is this material compatible?'), the supply to the bottleneck and later the output from the bottleneck will back up. Blockage of the system ("starvation" upstream, 'blocking' downstream).
	Two-Phase Heuristic for Disassembly Scheduling	Kim et al. (2007) https://doi.org/10.1080/00207540500244443 https://doi.org/10.1243/09544054JEM349	**2**	Only indirectly taken into account. The model works with fixed parts lists and fixed disassembly plans; commonality (shared parts) is taken into account, but not the fact that individual cores require additional or alternative steps. If the process planning phase has already determined the optimal disassembly depth or sequence for a specific core quality, these 'quality decisions' are incorporated into the scheduling step model as fixed parameters. The model therefore schedules what process planning has defined as standard based on (possibly variable) qualities. One could say that the stochastics are not calculated within the algorithm, but have been 'frozen' in the input parameters (structure and yield factors).	**1**	If so, then only based on conservatively assumed planned quantities. Kim et al. model the disassembly quantities deterministically; random scrap rates are not explicitly represented stochastically. Later work (e.g. capacitated disassembly scheduling under stochastic yield) shows that real yield uncertainty leads to significant cost effects that are not reflected in a deterministic heuristic such as Kim's Two-Phase. https://doi.org/10.1016/j.ejor.2017.08.032	**1**	Not addressed at all. Disassembly times are deterministic parameters; the heuristic optimises across quantities and sequences, not across distributed times.	**1**	Not explicitly taken into account. The model assumes that all planned dismantling steps are technically feasible and can be carried out immediately. Additional, stochastic waiting times are not modelled. The heuristic therefore systematically produces overly optimistic throughput times.
	Shifting Bottleneck Heuristic	Yu & Lee (2018) https://doi.org/10.1016/j.cie.2018.04.048	**3**	Different reprocessing routes and machine selection are represented in the model (flexible job shop). The stochastic nature of the routes (i.e. the fact that it is only clear after diagnosis whether route 1 or route 2 is necessary) is not modelled dynamically. The model works with routes that are known a priori. In a flexible job shop, operations could be performed on different alternative machines. In Yu and Lee's model, however, each operation on a component is assigned to a pre-specified workstation. Flexibility in the sense of variable routes for the same part is therefore not part of the mathematical model.	**1**	The paper assumes deterministic quantities of returning cores and solves the matching problem between recovered components and remanufactured products.	**1**	Yu & Lee work with deterministic processing times per machine and work step, as is customary in classic scheduling literature. Typical reman dispersion (e.g. 'slightly dirty' vs. 'heavily corroded') is not modelled probabilistically.	**3**	This is precisely where heuristics come into their strength: they model component matching requirements – certain components may only be combined with certain product instances. This is a form of technical incompatibility that is stored in the mathematical model as constraints. Unplanned, situational incompatibilities (e.g. clarification cases in quality management, missing documentation) are not modelled. The model perfectly masters 'planned incompatibility' (this part MUST go with that housing) through its family logic. However, it is blind to 'unplanned incompatibilities' (this part CAN SUDDENLY NOT be used), as it assumes perfect, trouble-free execution of each plan step.
Meta-Heuristic	Integrated scheduling for remanufacturing system considering component commonality	Guo et al. (2023) https://doi.org/10.1016/j.cie.2023.109419	**3**	The model explicitly maps several stages (disassembly, reprocessing, reassembly) and routing alternatives via commonality (parts can be reused in different products). The route logic is parameterised deterministically; quality-related additional steps or dynamic route changes during execution are not modelled as random variables, but via fixed process plans. The model simulates a scenario in which assembly is not blocked simply because a specific original part is still stuck in cleaning. The 'additional quality-related steps' mentioned in your assessment are missing. Each part of a type goes through exactly the same predefined sequence of operations.	**1**	The authors focus on the use of already reprocessed components and commonality, not on scrap probabilities or complete core failure. Random yield is taken into account at most implicitly via conservative resource/part availability assumptions; true yield stochastics remain unmodelled.	**1**	Processing times are used as deterministic parameters; multi-criteria optimisation (makespan, energy) makes the plan robust against peak loads, but not against explicit time stochastics.	**4**	The paper explicitly addresses component commonality and matching – i.e. which parts are compatible with which products. These technical compatibility conditions are firmly anchored in the model and are respected by the GA. Situational incompatibilities (e.g. external approvals, detailed problems with individual cores) can only be mapped if they are known in advance as model restrictions.
	Optimization of remanufacturing flexible job shop scheduling	Chen & Gao (2020) 10.25236/AJETS.2020.030809	**1**	The problem is a flexible job shop – alternative machines for each operation, but the sequence of operations per job is fixed. Machine flexibility is modelled, but route stochastics in the sense of additional or alternative process steps (e.g. extra repair after diagnosis) are not part of the model.	**1**	The number of orders is considered a given; random yield or failure of individual cores is not explicitly treated as a random variable.	**4**	Uncertain processing times are modelled using triangular and trapezoidal fuzzy numbers; the GA optimises based on these fuzzy operation times and fuzzy due dates.	**3**	Machine capabilities and, if applicable, setups are taken into account as constraints; MSOS (Machine Selection and Operation Sequence) coding ensures that only permissible machines/operations are selected. Specific reman incompatibilities (e.g. only certain cores with certain tools) can only be mapped if they are programmed into these constraints; spontaneous/unclear transition issues are not taken into account.
	Scheduling in a three-stage remanufacturing system with machine blockage, deterioration and maintenance	Luo et al. (2025) https://doi.org/10.1016/j.swevo.2025.102185	**2**	The model depicts a three-stage structure (disassembly – reprocessing – reassembly); within the stages, blockages and maintenance effects are used, not alternative process routes.	**1**	The work focuses on machine status (blockages, deterioration, RMAs) and treats cores more as deterministic jobs.	**4**	A deterioration model is presented in which processing times become longer over time as the machines wear out; rate-modifying activities (RMAs, a type of maintenance) shorten the times again. The model explicitly depicts a realistic, time-dependent variance component of processing times; the metaheuristic (MMBO) optimises start times and maintenance times simultaneously in order to minimise the makespan in the event of blockages and wear. The authors expressly emphasise that the study takes place in a 'deterministic environment'. Random fluctuations in processing time (e.g. because a core is unexpectedly severely damaged) are not modelled as probabilistic variables; this is marked as 'future work'.	**3**	Primarily via blockage models (limited buffers, blocking downstream stages) and maintenance constraints. Transition problems resulting from buffer restrictions and maintenance (e.g. parts cannot be transported further) are explicitly mapped. Other technical incompatibilities (special tools, approvals) are also only taken into account here if they are formulated as constraints in the model.
Deterministic / optimal **Deterministische Optimierung (LP / MILP)**	More mixed-integer linear programming models	Wang et al. (2024) 10.1016/j.cie.2024.110379	**1**	Wang et al. develop MILP models for a deterministic three-stage scheduling problem (Disassembly–Reprocessing–Reassembly). The models assume fixed process routes and do not account for the fact that varying core conditions lead to different routing decisions. Stochasticity within the routes themselves is not part of the model.	**1**	Wang et al. develop MILP models for a deterministic three-stage scheduling problem (Disassembly–Reprocessing–Reassembly). The models assume fixed process routes and do not account for the fact that varying core conditions lead to different routing decisions. Stochasticity within the routes themselves is not part of the model.	**2**	MILP models for scheduling can theoretically capture variable processing times in the parameters (e.g., different disassembly/reprocessing/reassembly times per job). However, it is unclear whether Wang et al. explicitly work with stochastic or variable times or whether processing times are deterministic. The focus lies on the scheduling problem itself, not on modeling the root causes of variability (core quality, unexpected damage).	**1**	The model focuses on temporal planning of three standardized process steps and does not address technical incompatibilities between components, design variants, or alternative procurement paths that would result in variable transitions.
	Batch production scheduling problem in a reconfigurable hybrid manufacturing-remanufacturing system	Vahedi-Nouri et al. (2025) 10.1016/j.cie.2025.111099	**1**	Vahedi-Nouri et al. focus on batch production scheduling in a Hybrid Manufacturing-Remanufacturing System (HMRS) with reconfigurable machines. The study develops MILP and CP models for scheduling EOL products alongside new manufacturing, but does not model stochastic or quality-dependent process routes. The problem formulation assumes that products follow predefined routes without considering how varying core conditions lead to different routing decisions.	**1**	The paper addresses scheduling optimization in an HMRS context but does not explicitly model uncertainty in core failure rates or remanufacturability. While the motivation mentions EOL products, the scheduling models treat jobs deterministically without incorporating random failure probabilities or variable remanufacturability levels of individual cores.	**2**	The batch scheduling model on non-identical parallel reconfigurable machines can theoretically accommodate different processing times per job or batch. However, the paper does not explicitly address variable processing times driven by core quality uncertainty or unexpected damage during remanufacturing operations. The focus is on deterministic scheduling optimization rather than modeling the root causes of processing time variability in remanufacturing contexts.	**2**	The study explicitly considers "non-identical parallel reconfigurable machines" and batch grouping, which implies some consideration of machine-specific capabilities and setup requirements. Reconfigurable Manufacturing Systems (RMS) are designed to provide customized production capabilities and flexibility, which may indirectly relate to handling different product variants. However, the paper does not explicitly model technical incompatibilities between components, design versions, or variable transition times caused by engineering clarifications or part substitutions.
	A capacitated disassembly scheduling problem considering processing technology selection and parts commonality	Darghouth and Abdel-Aal (2021) 10.1007/s13243-021-00104-3	**2**	Darghouth and Abdel-Aal (2021) address a capacitated disassembly scheduling problem considering processing technology selection and parts commonality. The inclusion of "processing technology selection" suggests that different cores or components may require different processing paths or technologies, which indirectly relates to quality-dependent routing. However, the paper does not explicitly model stochastic or quality-driven process routes where the routing decision is a random outcome based on inspected core condition. The technology selection appears to be a deterministic optimization decision rather than a stochastic routing outcome driven by fluctuating core quality.	**4**	The paper explicitly considers stochastic yield in the disassembly process, which directly relates to uncertainty in core failure and component recoverability. Related work by the same authors on "Capacitated disassembly scheduling under stochastic yield" indicates that yield uncertainty—the unpredictable proportion of usable components retrieved from disassembly—is a central concern. This aligns with t of "arbitrary failure rates," as yield variability reflects the fact that not all cores yield the same number of reusable parts due to varying degradation and failure patterns. However, the formulation typically treats yield as a probabilistic parameter rather than explicitly modeling individual core failure mechanisms.	**2**	The paper focuses on capacitated disassembly scheduling, which inherently requires modeling of processing times to determine feasible schedules. Variable processing times driven by core condition variability are a common feature in disassembly scheduling literature. However, it is unclear from the available abstract and citations whether Darghouth and Abdel-Aal (2021) explicitly model stochastic or variable processing times, or whether they treat processing times as deterministic parameters. The related work on "Capacitated disassembly scheduling with random demand and operation time" suggests that variable operation times are addressed in the broader research program, but specific treatment in this 2021 paper requires verification.	**1**	The paper explicitly considers parts commonality, which relates to the fact that multiple end-of-life product types may share common components. This consideration implicitly acknowledges that different product variants, design versions, or generations may have compatibility issues during disassembly and reassembly. However, the paper does not explicitly model "variable transition times" caused by technical incompatibilities, engineering clarifications, or part substitution requirements. The focus appears to be on optimizing disassembly schedules given parts commonality, rather than modeling the time variability induced by incompatibility resolution.
Stochastische/ dynamische Optimierung	Stochastic dynamic programming	Lage Junior & Filho (2017) https://doi.org/10.1007/s10100-015-0428-1	**5**	Disassembly routings are explicitly part of the model: a core can run through different disassembly paths with different probabilities (e.g. depending on its state).	**4**	Random yield is linked to stochastic routes: certain routes end earlier (e.g. rejects) or deliver fewer parts. The heuristic recalibrates planning at each horizon roll based on the actual routes and quantities that occurred. Although the authors do not model 'real' failure rates, their mathematical approach (stochastic dynamic programming) is perfectly suited to integrating this challenge.	**2**	The primary work focuses on stochastic routes and quantities; processing times are assumed to be deterministic or fixed parameters. Variance in times is not modelled as a random variable, but would have to be incorporated into the parameters again via safety margins. At the period level, however, the rolling horizon concept has a dampening effect: outlier times in one period are 'priced in' during the next replanning cycle.	**1**	The model focuses on disassembly decisions, not on all technical details of the reprocessing or assembly stage.
	Stochastic model predictive control for remanufacturing system management	Li et al. (2021) https://doi.org/10.1016/j.jmsy.2021.02.002	**5**	Li et al. (2021) propose a stochastic model predictive control (SMPC) approach for remanufacturing system management that explicitly considers "high uncertainty in the product quality." The paper addresses uncertainty in core quality as a central modeling challenge, which directly drives variability in process routing decisions. The stochastic control framework is designed to manage systems where incoming core quality varies significantly, leading to different remanufacturing paths.	**3**	The paper explicitly models "high uncertainty in the product quality and system dynamics" in remanufacturing operations. While not framed specifically as "arbitrary failure rates," quality uncertainty in remanufacturing inherently encompasses variability in core failure states and remanufacturability. The stochastic model predictive control approach is designed to handle unpredictable core conditions, which implicitly includes varying failure rates. However, the paper does not explicitly model individual component failure probabilities as a separate parameter.	**3**	The SMPC approach is designed to manage "remanufacturing system management" under uncertainty, and variable processing times are a natural consequence of quality variation in remanufacturing operations. The paper focuses on optimizing "remanufacturing efficiency" in the presence of uncertainty, which typically requires modeling variable processing times. However, based on available abstracts, it is unclear whether variable processing times are explicitly modeled as stochastic parameters or emerge implicitly through the system dynamics model	**1**	The paper focuses on stochastic control and quality uncertainty but does not explicitly address technical incompatibilities between components, design versions, or product generations. There is no indication that the model captures variable transition times caused by part incompatibilities, engineering clarifications, or component substitution requirements. The scope appears limited to quality-driven uncertainty rather than design-related compatibility issues
	Scheduling a Stochastic Remanufacturing Process with Disassembly, Reprocessing and Reassembly	Fu et al. (2020) 10.1109/ICNSC48988.2020.9238106	**3**	Fu et al. (2020) address scheduling for "a stochastic remanufacturing process including three subsystems, i.e., disassembly, reprocessing and reassembly." The paper models routing decisions at the workstation level (e.g., which parallel disassembly or reassembly workstation a core is assigned to) under uncertainty. However, this is not fine-grained quality-dependent process routing within individual work systems. The model does not capture detailed intra-station routing decisions such as: "Does this core require cleaning → inspection → partial disassembly, or can it skip cleaning?" The stochasticity relates to workstation assignment and sequencing rather than detailed process-step selection driven by inspected core quality. Therefore, while routing variability exists, it operates at a higher abstraction level than the detailed, quality-driven process routes characteristic of remanufacturing operations at the work-system level.	**4**	The paper models a "stochastic multi-objective integrated disassembly" problem, and related work by the same research group explicitly addresses "stochastic yield" in disassembly processes. While Fu et al. (2020) do not explicitly use the term "failure rates," the stochastic nature of the remanufacturing process inherently captures uncertainty in which cores and components are recoverable and usable. The three-stage structure (disassembly, reprocessing, reassembly) suggests consideration of yield variability, though explicit modeling of failure rates requires verification from the full paper.	**3**	The paper addresses "Scheduling a Stochastic Remanufacturing Process," and scheduling problems at the workstation level inherently require modeling of processing times. Variable or stochastic processing times are likely modeled for workstation operations (e.g., total time at a disassembly workstation). However, this likely does not capture the fine-grained variability driven by individual core conditions—such as unexpected damage requiring additional cleaning, rework loops, or unplanned inspection steps within a workstation. The processing time variability is probably captured as stochastic parameters at the job/workstation level rather than modeled as dynamic, condition-dependent task sequences.	**1**	The paper focuses on scheduling optimization under stochastic conditions for the three-stage remanufacturing process (disassembly, reprocessing, reassembly). There is no indication that technical incompatibilities between components, design versions, or product generations are considered. The focus appears to be on operational uncertainty and scheduling efficiency rather than design-related compatibility issues or variable transition times caused by part substitution or engineering clarifications.
AI-based	Deep reinforcement learning for real-time scheduling of collaborative customization remanufacturing	Yazdanparast et al. (2025) https://doi.org/10.1016/j.rcim.2025.102980	**5**	Mapped across the state space: quality type, necessary rework operations and intermediate buffers are state variables; the agent learns how different paths affect system utilisation and future start times. Only the route variants stored in the simulation model are learned; completely new error patterns require retraining.	**4**	In the simulated environment, rejects and cancellations occur as events; the reward penalises late or unfulfilled orders, so that the agent learns policies that select buffers and sequences in such a way that failure risks are cushioned in terms of timing. The policy is always linked to the learned yield distributions; significant changes in distribution lead to mismatches.	**4**	Variability in processing times (disruptions, rework) is part of the environment; the agent learns how different implementations shift its planned end times and learns, for example, to make more robust start decisions (not 'on the edge'). There is no explicit model of distribution; robustness arises empirically through experience, not analytically.	**3**	Known incompatibilities (certain combinations of parts/machines, approvals) are coded in the MDP as impermissible actions or states; the agent then only generates technically permissible start time decisions. Spontaneous, unmodelled incompatibilities are not taken into account – human intervention is still required here.
	Lead time prediction using machine learning algorithms	Lingitz et al. (2018) https://doi.org/10.1016/j.procir.2018.03.148	**4**	If routing variants (e.g. additional rework step) are reflected in the input characteristics (product condition, process path), the model 'learns' their influence on the actual throughput time. The route itself is not optimised, only its time sequence; training data is missing for new or rare routes.	**3**	Failures are not typically included in the training as 'finished LT'; predominantly successful runs are modelled. ML-LT prediction therefore primarily addresses time variance, not the probability that an order will fail completely. The model excels at cushioning the 'quality loss' of parts through ML-driven alternative strategies (use of new parts). However, since it constructs a system without 'real failure' (cancellation), addressing failure rates is actually limited to variance control.	**5**	The ML models implicitly integrate processing, waiting and downtime into the LT forecast and can also provide quantiles (e.g. 80th percentile) instead of a fixed value. For lead time scheduling, this means that start/end dates are based on data-driven, variability-sensitive LTs instead of rigid master data.	**3**	If certain combinations (material, machine, process variant) systematically lead to longer lead times and are coded in the features, the model captures this effect. Ad hoc incompatibilities that occur rarely are hardly learned due to the small number of cases. In addition, technical barriers are less likely to be picked up due to the simplification of the hardware side (capabilities of individual machines).
	Reinforcement learning and digital twin-based real-time scheduling method in intelligent manufacturing systems	Zhang et al. (2022) https://doi.org/10.1016/j.ifacol.2022.09.413	**4**	The DT can model multiple paths and their probabilities; the DRL agent 'experiences' in the simulation how different route choices lead to different end times. Modelling effort is high; new/unplanned routes do not initially exist in the DT.	**5**	Yield uncertainty can be explicitly mapped stochastically in the DT; the agent learns scheduling strategies that deliver acceptable adherence to deadlines despite rejects (e.g. earlier starts, parallel dismantling). Quality depends heavily on the calibration of the failure distributions in the DT.	**5**	DT models typically contain distribution-based process times; DRL learns policies that deal with this variance, e.g. by starting critical jobs early or loading certain machines differently. In contrast to purely data-driven DRL approaches, process engineers can explicitly control the time distributions in the DT.	**5**	DT-DRL addresses transition times and technical incompatibilities, provided these are encoded in the digital twin as sequence-, buffer- or state-dependent transition/set-up times; the RL agent optimises its start time decisions precisely against these (variable) transition dynamics.