Contents lists available at ScienceDirect

Robotics and Computer Integrated Manufacturing

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

Full length Article

Petri net-based scheduling strategy and energy modeling for the cylinder block remanufacturing under uncertainty

Shitong Peng[a,d], Tao Li[a,*], Jiali Zhao[b], Yanchun Guo[a], Shengping Lv[c], George Z. Tan[d], Hongchao Zhang[a,d]

[a] *Institute of Sustainable Design and Manufacturing, Dalian University of Technology, Dalian, China*
[b] *School of Mechanical & Electronical Engineering, Lanzhou University of Technology, Lanzhou, China*
[c] *College of Engineering, South China Agricultural University, Guangzhou, China*
[d] *Department of Industrial, Manufacturing, & Systems Engineering, Texas Tech University, Lubbock, TX, USA*

ARTICLE INFO

Keywords:
Remanufacturing scheduling
Petri net
A* algorithm
Engine remanufacturing
Remanufacturing uncertainty

ABSTRACT

Scheduling has been extensively applied to remanufacturing for the organization of production activities, and it would directly influence the overall performance of the remanufacturing system. Since the conjunction of Petri net (PN) and artificial intelligence (AI) searching technique was demonstrated to be a promising approach to solve the scheduling problems in manufacturing systems, this study built a transition timed PN combined with heuristic A* algorithm to deal with the scheduling in remanufacturing. The PN was applied to the formulation of remanufacturing process, while the A* algorithm generated and searched for an optimal or near optimal feasible schedule through the reachability graph (RG). We took the high value-added cylinder block of engine as a research object to minimize the makespan of reprocessing a batch used components. This scheduling problem involved in batch and parallel processing machines, and the uncertain processing time and routes will complicate the scheduling problem. Three heuristics were designed to guide the search process through the RG in PN. To avoid state space explosion and select promising nodes, a new rule-based dynamic window was developed to improve the efficiency of the algorithm, and this rule was examined to outperform the conventional one. Under the determined scheduling strategy, the dynamic behavior of energy consumption rate during the processing time was simulated using PN tool, which would assist remanufacturers to develop potential strategies for energy efficiency improvement. Considering the uncertainty of processing time, the Monte Carlo simulation method was adopted to statistically analyze the distributions of makespan and total energy consumption, which would contribute to the comprehensive production scheduling and energy profile assessment for sustainable remanufacturing.

1. Introduction

The rapid development of vehicle industry in China enables a higher living standard national-wide, while simultaneously brings issues like ecological degradation and energy consumption. As a new manufacturing paradigm, remanufacturing has been a rising industry with remarkable potentials and widely recognized as an outstanding end-of-life (EOL) alternative to extend the life cycle of products [1–3]. Based on the report from China Automotive Technology and Research Center (CATARC), the total amount of scrapped vehicles in 2020 is projected to be 99.5 million [4]. These substantial decommissioned vehicles are expected to fulfill the requirement of remanufacturing sources and enlarge the production scale of engine remanufacturing.

Different from conventional manufacturing, remanufacturing takes the scrapped products as workblank, which results in higher inherent uncertainties of the remanufacturing system. The decommissioned products experienced varying operation conditions during their service lives. Thus, a batch of returned products usually has diverse damage types and degrees. The stochastic returns and quality variations would lead to uncertainties in the remanufacturing process, such as operation time, reprocessing routes, and remanufacturing cost. Guide et al. [5] had confirmed that, compared with the traditional manufacturing, the planning and controlling of remanufacturing system are more complex because of the higher degree of variability. Additionally, in a typical manufacturing process, one machine is allowed to process one component at a time, while the remanufacturing cleaning equipment could

[*] Corresponding author.
E-mail addresses: lt_dlut@163.com, litao@dlut.edu.cn (T. Li).

https://doi.org/10.1016/j.rcim.2019.03.004
Received 24 July 2018; Received in revised form 25 February 2019; Accepted 13 March 2019
Available online 19 March 2019

Acronyms

AI	Artificial Intelligence
CATARC	China Automotive Technology and Research Center
DES	Discrete Event System
DWS	Dynamic Window Search
ECR	Energy Consumption Rate
EOL	End-of-Life
FIFO	first in and first out
FMS	Flexible Manufacturing System
GA	Genetic Algorithm
HDR	Heuristic Dispatching Rules
IEA	International Energy Agency
LPT	Longest Processing Time
MC	Monte Carlo
MPRT	Maximal Perfect Resource-Transition
PN	Petri net
PRT	Perfect Resource-Transition
PTPN	Place-timed Petri net
RG	Reachability graph
SPT	Shortest Processing Time
TTPN	Transition-timed Petri net

clean multiple parts simultaneously, which complicates production modeling and scheduling. Current shop scheduling problems mainly focused on conventional manufacturing, devoid of the specific uncertainty and operation characteristic of remanufacturing. Although the production schedule applied to remanufacturing cases have been addressed by [6–8], they mainly involved in the optimization of completion time without the consideration of energy consumption behavior under uncertainty. Similarly, their algorithms solving remanufacturing scheduling problems are also the intelligent algorithms such as the artificial bee colony algorithm and genetic algorithm (GA), which is exclusively appropriate for off-line scheduling due to the time-consuming computation process, particularly solving large-scale problems. Therefore, efficient approaches are desirable for rapidly solving scheduling problems in the remanufacturing system.

For a specific used component, it generally has multiple remanufacturing routes due to varying damage types and degree. Additionally, operations of the remanufacturing process could be processed on parallel machines (machine flexibility), which forms a typical flexible manufacturing system (FMS). Since FMS could be regarded as a discrete event system (DES) including loading, unloading, processing, and machine failures [9], the Petri net (PN), as a graphical tool and mathematical formalism, can be an effective option to deal with the modeling, analysis, and scheduling of remanufacturing system. For example, the constraint that a machine processes multiple-components at a time significantly increases the difficulty in conventional scheduling model, while it could be easily described and embedded in scheduling through PN representation. PN-based scheduling has been widely performed on diverse FMS including the semiconductor manufacturing [10], crude oil operation [11], and automobile manufacturing [12]. Nonetheless, these studies were limited in conventional manufacturing field.

The development of remanufacturing inevitably demands tremendous energy consumption. Currently, energy modeling has been a hotspot in the field of sustainable manufacturing. Energy simulation allows the intuitive display of dynamic consumption behaviors and examination of the effectiveness of energy conservation techniques. For the flexible manufacturing systems, their energy consumption evaluation and analysis are more complex. Additionally, the uncertainties in manufacturing system would complicate the energy modeling. Previous studies [13–15] have developed simulation methods to present and characterize the dynamic energy or emission flows of the conventional manufacturing system. However, these works merely examined the energy or emission flows in conventional manufacturing and failed to consider uncertain factors. With the scheduling scheme, PN technique enables swift and holistic simulation of energy consumption under uncertain environment. Systematic development and analysis of energy consumption model are of great importance in manufacturing to understand realistic consumption behaviors and identify potential energy reduction opportunities.

The literature review indicates that studies [16–19] applied PN to remanufacturing practice were primarily for the disassembly process. Few researches have incorporated the PN into the modeling and scheduling remanufacturing process. We developed a novel stochastic PN method to model the remanufacturing of cylinder block, which is a prerequisite of the scheduling with the objective of makespan minimization. Since the flexible job-shop scheduling problem (FJSP) belongs to an NP-hard combinatorial optimization problem, A* search technique based heuristic search algorithm was adopted to seek the optimal or near-optimal solution, i.e., the firing sequence in reachability graph (RG) of stochastic PN. Compared with the intelligent

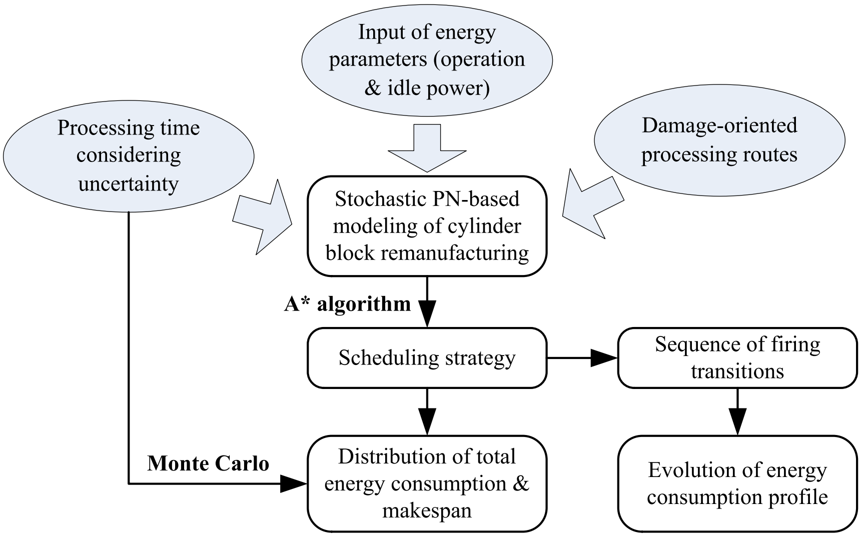

Input of energy parameters (operation & idle power)
Processing time considering uncertainty
Damage-oriented processing routes
Stochastic PN-based modeling of cylinder block remanufacturing
A* algorithm
Scheduling strategy
Sequence of firing transitions
Monte Carlo
Distribution of total energy consumption & makespan
Evolution of energy consumption profile

Fig. 1. A brief illustration of the undertaken work.

algorithms, this algorithm enables more efficient searching and is appropriate for real-time scheduling without much compromising of the accuracy of solution [20]. In this study, we merely concerned the makespan in the determination of scheduling scheme. According to the onsite survey at SINOTRUK, Jinan Fuqiang Power Corp. Ltd., a large engine remanufacturer in China, the processing time of each operation in cylinder block remanufacturing is statistically presented in the form of triangular distribution. Under the scheduling strategy and the uncertain processing time, the energy consumption behavior during the processing period was simulated with PN model. The Monte Carlo (MC) simulation results concerning total energy consumption and completion time were statistically analyzed to figure out their distributions under uncertainty. A brief description of the present study is shown in Fig. 1.

In this study, the stochastic PN modeling, scheduling, and analysis were performed through the remanufacturing case of cylinder block. Amongst the primary components of engine, cylinder bock has the largest mass, highest price, and complicated structure. Its remanufacturing process is high value-added and results in significant material and energy savings. The minimization of makespan implies higher production efficiency and the energy modeling reveals the evolution of energy consumption process of entire cylinder remanufacturing process, which would benefit the comprehensive production scheduling and energy profile assessment for sustainable remanufacturing. The contributions lie in the following points: (1) to the best of our knowledge, it is the first attempt to apply the stochastic PN to the modeling and scheduling of remanufacturing process; (2) the transition timed PN of FMS rather than place timed PN in previous studies would avoid deadlocks during the dynamic variation of markings; (3) different from traditional PN using a pair-wise place and token to represent a machine tool, the net in the present study employs distinguishable tokens in a place to represent the parallel machines, which would simplify the model without compromising its capability of formalized system description; (4) a heuristic A* search algorithm is developed to efficiently obtain the scheduling scheme, and a new rule-based window search technique is embedded in the algorithm to shrink search space; and (5) based on the processing parameters collected from the real-world case, the total energy consumption and makespan under uncertainty are obtained through MC simulation.

2. Fundamental concepts and definitions

2.1. Basic definitions of PN

A PN is a bipartite, weighted, directed graph comprised of places, transitions, and arcs that connect place and transition, either from transition to place or the opposite [21]. It can be presented by a five-tuple $PN = (P, T, F, W, M_0)$, where the P and T denote the finite and disjoint set of places and transitions, respectively, namely, $P \cap T = \varnothing$, $P \neq \varnothing$, $T \neq \varnothing$. F refers to the set of directed arcs, $F \subseteq (P \times T) \cup (T \times P)$. W and M_0 are the weighting function and initial marking, respectively. A k-weighted arc can be regarded as k parallel unit arcs and label of the unit arc is generally omitted. The marking or state of PN could be denoted by $M = [M(p_1), M(p_2), \ldots, M(p_m)]$, in which the m-th element $M(p_m)$ means the number of tokens in the p_m place. A five-tuple PN would be presented as $PN = (N, M_0)$, where $N = (P, T, F, W)$ refers to PN structure without initial marking. For $t \in T$, ${}^{\cdot}t(t^{\cdot})$ means the set of pre-places (post-places) of the transition t. For $p \in P$, ${}^{\cdot}p(p^{\cdot})$ means the set of pre-transitions (post-transitions) of the place p.

A transition $t \in T$ is enabled or fireable if $\forall p \in {}^{\cdot}t$: $M(p) \geq w(p, t)$, and presented as $M[t >$. $w(p,t)$ denotes the weight of directed arc from p to t. Similarly, $w(t, p)$ means the weight of arc from t to p. Firing of an enabled transition would result in the generation of reachable marking M' from the marking M, and it could be presented as $M[t > M'$. The successive marking M' could be determined by Eq. (1). If $\exists\, M_i[t_i > M_{i+1}$, $i = 0, 1, 2, \ldots, k$, the path from M_0 to M_{k+1} can be presented by a sequential string $\alpha = t_1 t_2 t_3. ..t_k$, and this path can be called a circuit if $M_0 = M_{k+1}$. The set of all possible reachable markings of PN from M_0 is denoted as $R(N, M_0)$. A sequence of firings on enabled transitions results in a sequence of token distributions, i.e., the marking in a net, which can be completely presented by the RG. Therefore, dynamic behaviors of systems could be tracked by the RG of a PN [22]. In this transition enabling rule, the underlying assumption is that unlimited amounts of tokens can be accommodated in each place, which is also called infinite capacity PN. As opposed to the finite capacity net considering the upper limitation of tokens in each place, the infinite capacity net is more frequently used in FMS cases.

$$M'(p) = \begin{cases} M(p) - w(p, t) & \forall\, p \in {}^{\cdot}t \setminus t^{\cdot} \\ M(p) + w(t, p) & \forall\, p \in t^{\cdot} \setminus {}^{\cdot}t \\ M(p) + w(t, p) - w(p, t) & \forall\, p \in {}^{\cdot}t \cap t^{\cdot} \\ M(p) & \forall\, p \notin {}^{\cdot}t \cup t^{\cdot} \end{cases} \tag{1}$$

2.2. Structure of the timed PN

According to the time property adopted in the PN, the timed PN primarily includes place-timed PN (PTPN) and transition-timed PN (TTPN) [20]. The structure of TTPN is shown in Fig. 2(a). Each transition in the PN is associated with firing time. The tokens are taken from input places at the beginning of the firing time while deposited in the output places at the end of firing time, and the tokens can be supposed to remain "in" the transition during the firing time interval [23]. Different from the TTPN, each transition in PTPN is fired and completed instantly and time delay is associated with the places instead of transitions. The instantaneous transition can be regarded as a special transition with zero delay. As shown in Fig. 2(b), the instantaneous transition t_s fires immediately, and the token is considered to sojourn in the place $t_s^{\cdot}$ rather than in the transition for a delay time. Subsequently, the transition t_e fires to remove token in ${}^{\cdot}t_e$ and deposit tokens in $t_e^{\cdot}$. The

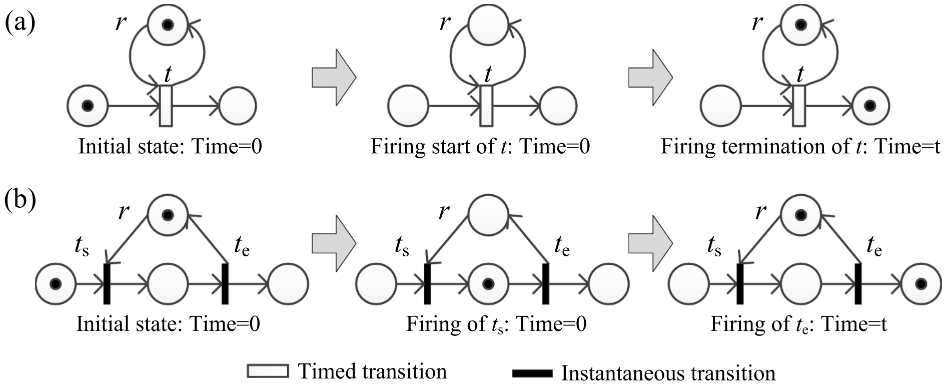


Fig. 2. Structure of PNs with different types of transitions: (a) transition-timed PN, (b) place-timed PN.

typical interpretations of places and transitions in modeling is that place represents condition and transition refers to event [21]. Explanations of TTPN and PTPN of a specific case are slightly different. Let us take the example of milling a part on the machine tool. In the TTPN of Fig. 2(a), place *r* with (without) a token indicates the milling machine is (not) available. The transition *t* represents the milling process. ${}^{\bullet}tr$ and $t^{\bullet}\backslash r$ mean the buffers holding the input components and milled components, respectively. In the PTPN of Fig. 2(b), both ${}^{\bullet}t_s\backslash r$ and $t_e{}^{\bullet}\backslash r$ indicate the buffer area for components. The place $t_s{}^{\bullet}$ represents the component is in processing. Transitions t_s and t_e denote the state changes from being held in buffer to beginning of processing and from processing to the buffer, respectively. For the PN of an FMS, places could be classified into operation places (${}^{(o)}t$ or $t^{(o)}$) and resource places (${}^{(r)}t$ or $t^{(r)}$).

Compared with the TTPN, PTPN has been more extensively employed in the modeling and scheduling of FMS such as the work of [10,22,24,25]. However, TTPN is more advantageous over the PTPN to refrain from deadlock. Xing et al. [26] characterized the deadlock in a PN model as a maximal perfect resource-transition (MPRT) circuit saturated at a reachable marking. The resource-transition circuit θ, as the name suggests, only contains transitions and resource places. In the example of an FMS shown in Fig. 3(a), the path $\theta = r_1t_{22}r_2t_{12}r_1$ belongs to a resource-transition circuit. Let $R[\theta]$ and $T[\theta]$ denote the sets of resource places and transitions in the circuit θ, respectively. A circuit θ is considered as perfect resource-transition (PRT) if $({}^{(o)}T[\theta])^{\bullet} = T[\theta]$. A PRT circuit is saturated if it satisfies the $M({}^{(o)}T[\theta]) = M_0(R[\theta])$. The theorem to identify the existence of deadlock is shown as follows.

Theorem 1. [26] *A system of simple sequential processes with resources is alive if and only if, at any reachable markings of (N, M_0), no PRT circuit is saturated.*

The initial marking of PTPN in Fig. 3(a) is $M_0 = 3p_{1s} + p_{2s} + 2r_1 + r_2$, and deadlock will occur at the reachable marking of $M = p_{1s} + 2p_{11} + p_{21}$ when the circuit θ is saturated, i.e., $M({}^{(o)}T[\theta]) = M_0(R[\theta]) = 3$. However, as shown in Fig. 3(b), the equivalent TTPN of the PTPN would avoid the deadlocks in terms of the occasions implied in Theorem 1. In the case (Fig. 3) of two operations performed on r_1 and r_2, the essential reason for getting stuck in deadlock is that release of the processing machine (token in *r*) requires the availability of next machine, while the resource would be released directly after the completion of operation.

2.3. Transition timed PN for scheduling

Let us suppose the FMS is comprised of *m* types of resources for the processing of *n* types of components. Each type of resource generally refers to a machine for processing one operation of parts. The resource set is denoted as $R = \{r_i|i = 1, 2, \ldots, m\}$, where r_i is the set of machines for the *i*th operation. This set can be presented by $r_i = \{r_{is}|s = 1, 2, \ldots, k\}$, where r_{is} means the serial number of machines for the *i*th operation. The capacity of r_i is marked as $C(r_i)$, namely, the amounts of parts processed simultaneously on one machine. Let the $Q = \{q_j| j = 1, 2, \ldots, n\}$ denote a component set, in which q_j is the set of type-*j* components to be processed. The type-*j* part has a specific processing route pr_j which can be expressed as a sequence of operations $pr_j = o_{j1}o_{j2}o_{j3} \ldots o_{jL(j)}$. o_{ji} means the *i*th operation in pr_j and L(*j*) denotes the length of pr_j. Each machine in charge of processing one specific operation, i.e., one operation requires only one resource (machine) and successive operations need different resources. Considering that different route may share the same operation, these identical operations emerge into one single operation. In the present study, processing route of type-*j* component is expressed by the firing sequence of transitions $\alpha_j = t_{j1}(r_{1,s1})t_{j2}(r_{2,s2}) \ldots t_{jL(j)}(r_{L(j),sL(j)})$, where the $r_{i,si}$ in parentheses indicates the serial number of machine for the transition t_{ji}. For simplicity, the resource place holding resource (machine) r_{is} is also denoted as r_i. Initial marking of this resource place is the length of r_i set, i.e., the number of machines in r_i. Let P_R and P_O represent the set of all resource places and operation places, respectively. For a marking $M_f \in R(N, M_0)$, if $\sum_j M_0({}^{\cdot}t_{j1}) = \sum_j M_f(t_{jL(j)}{}^{\cdot})$ and $M_f\ (r_i) = M_0\ (r_i)$, M_f is called the final marking of (*N*, M_0). If a sequence of transitions τ satisfy $M_0[\tau > M_f$, then τ is called a feasible schedule.

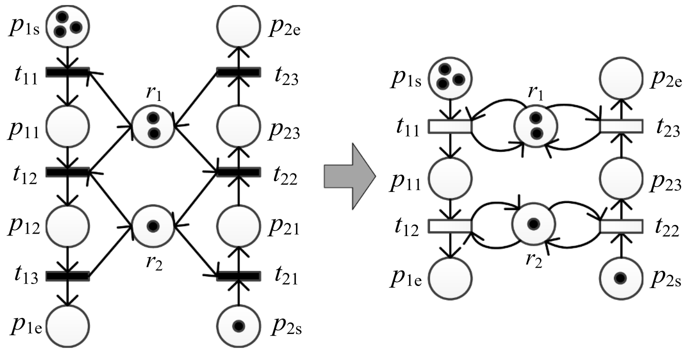
In the timed PN, delays with regard to transitions or places are selected deterministically. While delays in stochastic PN are randomly chosen under specified possibility distributions. It is assumed that the processing time of each operation or transition, denoted by $d(t(r))$, satisfies triangular distribution. The data regarding these distributions were mainly obtained from workers with abundant processing experience of over 5 years in Jinan Fuqiang Power Corp. Ltd.

3. System and problem description

3.1. Cylinder block remanufacturing

Seven primary components in an engine, also refer to “seven pieces,” include cylinder block, cylinder head, connecting rod, crankshaft, flywheel housing, gearbox, and flywheel. Engine remanufacturing mainly engages in the first six components, while the reassembly adopts a totally new flywheel rather than a remanufactured counterpart. The investigated cylinder block (Fig. 4) belongs to the WD615-87 Diesel Engine. Cylinder block possesses the largest weight amongst these major components, which necessitates the remanufacturing for material conservation and cost-saving.

The remanufacturing process generally includes disassembly, cleaning, testing, reprocessing, and reassembly. Reprocessing, however, suggests a process chain or route in which multiple machines and processes are performed jointly to recover the dimension and shape of used parts. Each type of component has a specific route for reprocessing. After the inspection of returned cylinder blocks, two processing routes are conducted for the slight and severe abrasion of spindle holes,



(a) PTPN (b) TTPN

Fig. 3. An illustrative example of FMS in the form of PTPN and TTPN.

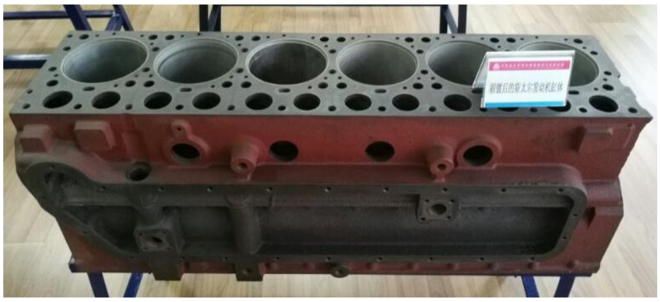


Fig. 4. The investigated cylinder block.

respectively, to recover the design dimensions ($108_{0}^{+0.022}$ mm). Details regarding the reprocessing of used cylinder blocks are displayed in Fig. 5. The severely damaged parts share all the operations the slightly damaged parts possess, and they have two additional operations (O_3 and O_4). Apart from the O_1, O_3, and O_4, other operations were processed on a single machine.

Detailed operations and relevant machines are presented in Table 1. Considering the variations of damage degree, parallel equipment, and workers' operation, the processing time of each operation would be varying during the whole processing period. As listed in Table 1, processing time is presented in the form of triangular distribution with three parameters: optimistic value, most plausible value, and pessimistic value. Since the O_1 and O_8 operations are performed manually, the rated powers of corresponding equipment are considered as 0. The power measurements of milling machine tool [27], turning lathe [28], and laser cladding equipment [29] indicated that the ratio of idle power to operation power is roughly 25%–70%. In the present study, we assume the idle powers of machines are 30% of the rated power due to the difficulty of accurate power measurement of all the machines. Time consumption of O_1, O_3, and O_4 are significantly greater than others, thus parallel machines are arranged for these operations.

The states of a machine usually include starting up, idle, processing, and shutdown. Due to the short duration of starting up and shutdown, energy consumption of these stages is neglected in the present study, which is consistent with prior works [13,30,31] regarding the determination of energy use. Therefore, energy consumption of a component is determined by the duration of operation and idleness as well as specific machines, as shown in Fig. 6. The area under solid line of power (P_{ri}^{o} and P_{ri}^{I}) indicates the total energy consumption of reprocessing a part. The reduction of the operation time (T_{ji}^{o}) and idle time (T_{ji}^{I}) will decline the processing time and energy consumption, which requires an appropriate scheme of machine selection and processing sequence. The variables of energy use and completion time are interrelated, and we would investigate the dynamic behavior of energy consumption under the scheduling strategy for makespan minimization.

3.2. Problem description

In this FJSP, the returned cylinder blocks are classified into two types: severely damaged (q_1) and slightly damaged parts (q_2), and the component set can be presented as $Q = \{q_1, q_2\}$. Their processing routes and the relevant optional machines are presented in Fig. 5. The common constraints of machines and jobs also applied in this scheduling include: (1) any jobs cannot be processed on multiple machines at a time, (2) an operation cannot be interrupted until the completion on the machine, (3) all the jobs have identical priority, and (4) all the machines are available at the beginning time. In the present study, we employed the case of remanufacturing three severely damaged and six slightly cylinder blocks and assume that the cleaning equipment can process three components at a time. Fig. 7 displays the PN model of cylinder block reprocessing in the form of TTPN. The p_{1s} and p_{2s} denote the starting place holding severely damaged and slightly damaged components, respectively, and p_e means place holding the completed parts. $\forall t \in T$, ${}^{(o)}t = t^{(o)} = 1$, ${}^{(r)}t = t^{(r)} = 1$. $q_1 = \{1, 2, 3\}$, $q_2 = \{4, 5, 6, 7, 8, 9\}$. $M_0(q_1)$ and $M_0(q_2)$ equal the length of q_1 and q_2. Different from conventional PN, we numbered and colored all the tokens in places of Fig. 7 and made them distinguishable. For $\forall M \in R(N, M_0)$, $\forall p \in P$, let $NM(p)$ represent the set of tokens of $M(p)$. Thus, $NM_0\ (p_{1s}) = q_1$, $NM_0\ (p_{2s}) = q_2$, $NM_0\ (r_1) = \{r_{11}, r_{12}, r_{13}\}$, $NM_0\ (r_3) = \{r_{31}, r_{32}\}$, $NM_0\ (r_4) = \{r_{41}, r_{42}\}$, $\forall i = 2, 5, 6, 7, 8, 9$, $NM_0\ (r_i) = \{r_i\}$. For $\forall f \in (P \times T) \cup (P \times T)$, $f \neq (p_{18}, t_{19}) \cup (t_{19}, p_e)$, $w(f) = 1$. $w(p_{18}, t_{19}) = w\ (t_{19}, p_e) = 3$.

The objective of this scheduling problem is to minimize the completion time of all the used cylinder blocks. More specifically, the completion time of the last job on machine r_9, as presented in Eq. (2):

$$\min \max\{T^F(j, r_9) | j \in q_1 \cup q_2\} \tag{2}$$

where $T^F(j, r_9)$ denotes the firing completed moment of the transition r_9, i.e., the completion time of the jth job on machine r_9. The completion moment of the last job can be presented as $\max\{T^F(j, r_9) | j \in q_1 \cup q_2\}$. Additional constraints regarding jobs and operations can be described by Eqs. (3)–(7):

$$T^E(j, r_{im}) + T_{j,im}^{o} \le T^E(j', r_{im}) \quad \forall\, j \text{ and } j' \in q_1 \cup q_2,\ r_{im} \in NM_0(r_i) \tag{3}$$

$$T^E(j, r_{im}) + T_{j,im}^{o} \le T^E\left(j, r_{i'm}\right) \quad \forall\, j \in q_1 \cup q_2,\ r_{im} \text{ and } r_{i'm} \in NM_0(r_i),\ i < i' \tag{4}$$

$$T^E(j, r_{im}) + T_{j,im}^{o} = T^F(j, r_{im}) \quad \forall\, j \in q_1 \cup q_2,\ r_{im} \in NM_0(r_i) \tag{5}$$

$$\sum_j x(j, r_{im}, T) = 0 \text{ or } \sum_j x(j, r_{im}) = 3, \quad \text{for } i = 9 \tag{6}$$

$$\sum_j x(j, r_{im}, T) \le 1, \quad \forall\, r_{im} \in NM_0(r_i) \text{ and } i \ne 9 \tag{7}$$

where $T^E(j, r_{im})$ and $T^F(j, r_{im})$ indicate the starting time and completion

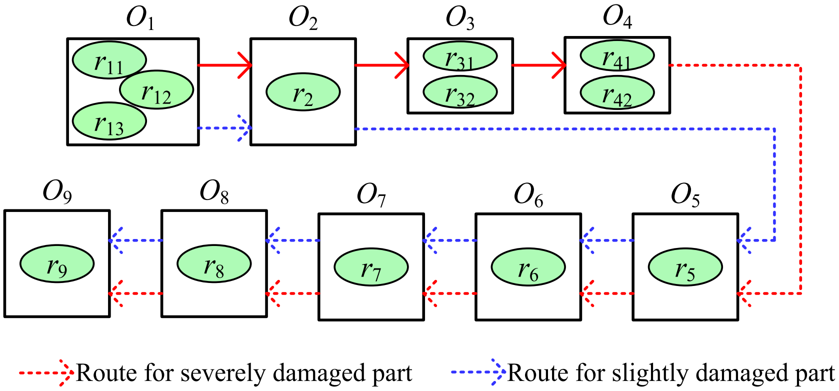


Fig. 5. Reprocessing routes of the cylinder block.

Table 1
Information on the operations and machines.

Serial number	Equipment	Operation	Rated power (kWh)	Time consumption (min)
r_{11}, r_{12}, r_{13}	Inspection benches	Test and install bowl-shaped plug	0	(69, 77, 86), (73, 80, 90), (79, 85, 94)
r_2	Surface grinding	Grind the crankcase	7	(42, 45, 48)
r_{31}, r_{32}	Spray machines	Recover dimensions	20	(90, 94, 98), (93, 100, 106)
r_{41}, r_{42}	Boring lathes	Bore bearing holes	1.1	(50, 56, 61), (56, 60, 67)
r_5	Honing lathe	Hone main bearing holes	1.6	(21, 25, 29)
r_6	Water detector	Air-leakage test	20	(22, 25, 28)
r_7	Press machine	Install cylinder sleeves	2.2	(9, 10, 11)
r_8	Bench	Install camshaft bush	0	(17, 20, 22)
r_9	Cleaning machine	Clean cylinder block and oil gallery	120	(28, 30, 32)

time of processing *j*th job on the machine r_{im}, respectively, $T^o_{j,im}$ is the operational duration of *j*th job processed on the machine r_{im}, $x(j, r_{im}, T)$ is a binary variable. When $x(j, r_{im}, T) = 1$, the *j*th job is being processed on the machine r_{im} at time *T*; otherwise, $x(j, r_{im}, T) = 0$. Constraint (3) specifies the precedence of two successive jobs on machine r_{im}. A new task (*j*th job) of a machine could only be started after the task (*j*th job) completion on this machine. Constraint (4) ensures that a new operation can be started after the completion of the prior operation. Operations would be processed continuously without interruption as shown in constraint (5). The cleaning machine (r_9) in the last operation is capable of processing three components simultaneously (constraint (6)), and other machines can process at most one single job at a time (constraint (7)). Considering the processing time is assumed to satisfy triangular distribution, we adopted the optimistic, most plausible, and pessimistic processing time values of each machine to determine the range of makespan.

Aforementioned energy consumption of remanufacturing these jobs primarily considers the operation part and idleness part by multiplying the relevant power and time duration. Admittedly, the calculation process is simple and inaccurate. However, this computation method could roughly determine energy use and has been extensively adopted in many prior studies [31–35]. The determination of energy-related variables is shown in Eq. (8)–(11):

$$E_{rate}(T) = \sum_i \sum_m \sum_j x(j, r_{im}, T)\cdot P^o_{im} + \sum_i \sum_m \sum_j P^I_{im}\cdot(1 - x(j, r_{im}, T))\cdot nm(r_i)\cdot ft(r_{im}) \tag{8}$$

$$nm(r_i) = \begin{cases} 1 & \cup_{n\le i} NM({}^{\cdot}({}^{\cdot}r_n)) \neq \varnothing \\ 0 & else \end{cases} \tag{9}$$

$$ft(r_{im}) = \begin{cases} 1 & FT(r_{im}) \geq 1 \\ 0 & FT(r_{im}) = 0 \end{cases} \tag{10}$$

$$E_{total}(T) = \int_0^T E_{rate}(T) \tag{11}$$

where P^o_{im} and P^I_{im} are the power of machine r_{im} at operational state and idle state; $E_{rate}(T)$ and $E_{total}(T)$ denote the energy consumption rate and total energy consumption of the reprocessing system at time *T; nm*(r_i) examines whether the machines (colored tokens) in resource place r_i should be shut down; *ft*(r_{im}) tests if the machine r_{im} have been started up; *FT*(r_{im}) records the number of jobs processed on machine r_{im}. Eq. (8) indicated that the entire energy power of the remanufacturing system at every moment considering the different states of machines. As shown in Eq. (11), the total energy consumption is the integral of total power with respect to time.

4. Heuristic A* search algorithm

Since the processing sequence reflected in PN is essentially based on the RG, the A* search algorithm has been applied to the PN-based scheduling problems [24,36,37] due to its superiority in pathfinding and graph traversal. The basic idea is to explore the state space by examining the successors of already-explored states. As a combination of a marking and the pertinent transition sequence, a vertex (*M, α*) denotes the state and transition log of (*N*, M_0), such that $M_0[\alpha > M$. Two vertexes (M_1, α_1) = (M_2, α_2) if and only if $M_1 = M_2$ and $\alpha_1 = \alpha_2$. Vertexes (*M*, α_1) and (*M*, α_2) indicate reaching the marking *M* from M_0 through different paths (α_1 and α_2). For the determination of the best transition or node to expand next, the heuristic function of processing time in A* algorithm is defined as follows.

Definition 1. For a vertex (M, α), the estimated makespan f(M, α) from initial marking M_0 to goal marking M_f could be determined by Eq. (12):

$$f(M, \alpha) = g(M, \alpha) + h(M, \alpha) \tag{12}$$

where *g*(*M, α*) is the time from M_0 to *M*, i.e., firing completion of the last transition in *α; h*(*M, α*) is the unconstrainted minimal time from *M* to M_f.

When A* algorithm compares and selects candidate nodes to expand at state *M*, their *g*(*M, α*) are identical and the evaluation function *f*(*M, α*) mainly depends on *h*(*M, α*). The design of heuristic function *h*(*M, α*) is critical and it would directly affect the performance of A* search algorithm. Calculation of *h*(*M, α*) is based on the matrix of minimal processing time defined below.

Definition 2. ∀*p* and *p'* ∈ P_O, *α* is a feasible transition path from *p* to *p'*.

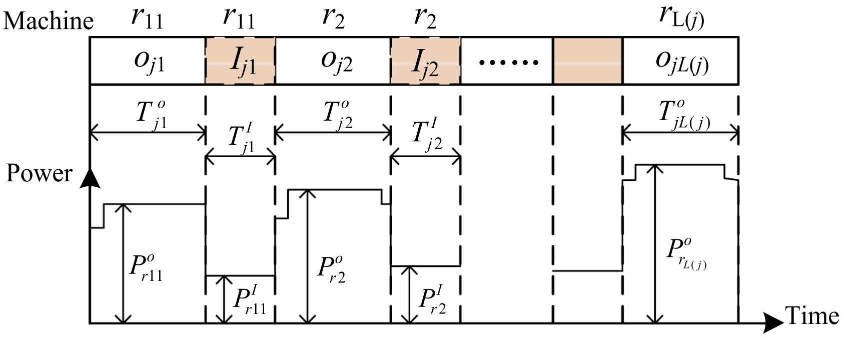


Fig. 6. Energy consumption rate during operations of the type-j component.

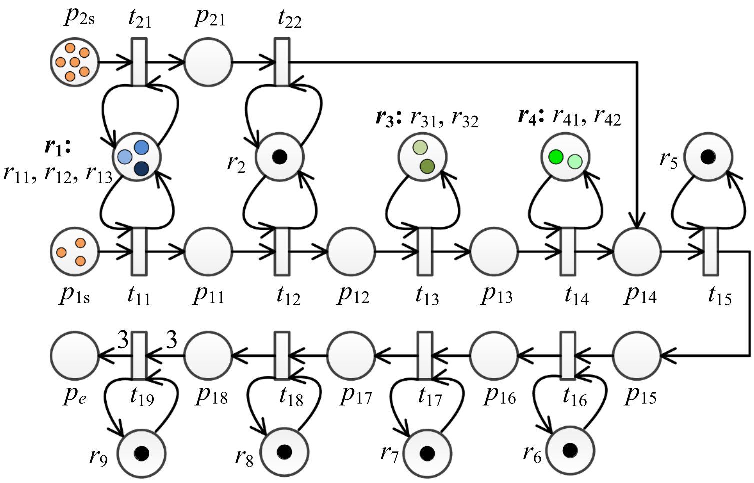


Fig. 7. PN model of cylinder block reprocessing.

Let $D(\alpha) = \sum_i d(t_i(r_{im}))$ denote the time from p to p' under path α, in which t_i (r_{im}) refers the transition between p to p', and $d(t_i$ $(r_{im}))$ is the delay of transition t_i under the machine r_{im}. The minimal processing time from p to p' is presented as $X(p, p') = \min\{D(\alpha)|$ for all $\alpha\}$. If the path from p to p' is unavailable, then $X(p, p') = \infty$. Therefore, $X(p, p')$ can be regarded as the entry of a matrix reflecting the minimal processing time between two operation places.

As the place p_e in Fig. 7 indicates completion of jobs, for $\forall p \in P_O$, X (p, p_e) is simplified as $X(p)$. To ensure that A* search algorithm can find the optimal solution, the heuristic function $h(M, \alpha)$ should be admissible [38]. For $\forall M \in R(N, M_0)$, it satisfies $0 < h(M, \alpha) \leq h^*(M, \alpha)$, in which denotes the actual minimum processing time from M to M_f, then $h(M, \alpha)$ is admissible, i.e., $h(M, \alpha)$ never overestimates the time to reach the goal marking. Three heuristic functions are proposed in this section.

Definition 3. For a vertex (M, α), a job $j \in q_1 \cup q_2$ begins its processing on machine r_{im} at time $g(M_1, \alpha_1)$ and with the operational duration of d $(t_i(r_{im}))$. Let $RT(j, r_{im})$ denote the remaining processing time of j on r_{im}, and $RT(j, r_{im}) = g(M_1, \alpha_1) + d(t_i(r_{im}))$-$g(M, \alpha)$. The total remaining operation time of the system (N, M_0), i.e., the gross sojourn time of jobs on machines, at the moment $g(M, \alpha)$ can be presented as $RT = \sum_i \sum_m \sum_j x(j, r_{im}, g(M, \alpha)) \cdot RT(j, r_{im})$, if the jth job is being processed on the machine r_{im} at time $g(M, \alpha)$, then $x(j, r_{im}, g(M, \alpha)) = 1$; otherwise, $x(j, r_{im}, g(M, \alpha)) = 0$.

Definition 4. Jobs at time $g(M, \alpha)$ are sorted by their states: held in a buffer or processed on a machine. $\forall p \in P_O$, define the total completion time of jobs (tokens) in place p as $\Sigma_p M(p) \cdot X(p)$, and the total residual time of jobs being processed on machine r_{im} after their completion is presented as $\Sigma_i \Sigma_m \Sigma_j x(j, r_{im}, g(M, \alpha)) \cdot X(t(r_i)^{\bullet})$. $t(r_i)$ is the relevant transition of r_i.

According to Definitions 3 and 4, the heuristic function $h_1(M, \alpha)$ can be expressed by Eq. (13) as below:

$$h_1(M, \alpha) = \sum_i \sum_m \sum_j x(j, r_{im}, g(M, \alpha)) \cdot (X(t(r_i)^{\cdot}) + RT(j, r_{im})) + \sum_p M(p) \cdot X(p) \tag{13}$$

Then, $h_1(M, \alpha)$ shows the unconstrained minimum residual time to complete all the jobs at time $g(M, \alpha)$. The underlying assumptions reflecting the unconstrained conditions mainly include: (1) all the machines are available at any time when jobs are scheduled to be processed, and (2) all jobs are processed along the path with minimal processing time, i.e., the most efficient machine is selected at each operation.

In heuristic function $h_1(M, \alpha)$, the determination of $X(p)$ completely disregards the waiting time of jobs for machines, or resources. To accurately estimate the time of completing a job, the second heuristic function $h_2(M, \alpha)$ would consider the waiting of jobs from $g(M, \alpha)$ to the possible beginning time of their next operations. The waiting time is defined and calculated as follows.

Definition 5. For a vertex (M, α), if $\exists$ a place r_i satisfies $M(r_i) = 0$ and M $({}^{\bullet}({}^{\bullet}r_i)) \neq 0$, then jobs in the place ${}^{\bullet}({}^{\bullet}r_i)$ are waiting to be processed on machine $r_{im} \in NM$ (r_i). Suppose the job j is being processed on r_{im} at time $g(M, \alpha)$, and let $l(j, r_{im}, g(M, \alpha))$ denotes the time r_{im} is available, i.e., releasing the job j. The minimum waiting time of jobs in ${}^{\bullet}({}^{\bullet}r_i)$ can be expressed as $WT_i = \min\{l(j, r_{im}, g(M, \alpha))$ - $g(M, \alpha)|\ r_{im} \in NM$ $(r_i)\}$. If M $(r_i) \neq 0$, then at least one machine in the place r_i is available at the time $g(M, \alpha)$, i.e., the waiting time $WT_i = 0$.

According to Definition 5, the second heuristic function $h_2(M, \alpha)$ is presented in Eq. (14):

$$h_2(M, \alpha) = \sum_i \sum_m \sum_j x(j, r_{im}, g(M, \alpha)) \cdot (X(t(r_i)^{\cdot}) + RT(j, r_{im})) + \sum_p M(p) \cdot X(p) + \sum_i (1 - \mathrm{sgn}(M(r_i))) \cdot WT_i \cdot M({}^{\cdot}({}^{\cdot}r_i)) \tag{14}$$

where sgn is a sign function. Similar to the $h_1(M, \alpha)$, the determination of $h_2(M, \alpha)$ also rests on the assumptions mentioned above. The primary difference is that $h_2(M, \alpha)$ takes one step further consideration of the waiting time of jobs before their next operations. Both $h_1(M, \alpha)$ and $h_2(M, \alpha)$ are admissible, and the proof is provided as follows.

Proof. For a vertex (M, α), the actual optimal processing sequence from M to M_f is denoted by γ^*. Let $d_1(j, \gamma^*, M, \alpha)$ and $d_2(j, \gamma^*, M, \alpha)$ be the total processing time and waiting time of jth job during the path of γ^*. The completion time of γ^* is presented as $l(\gamma^*)$, then it can be found that $l(\gamma^*)$ - $g(M, \alpha) = d_1(j, \gamma^*, M, \alpha) + d_2(j, \gamma^*, M, \alpha)$. $h^*(M, \alpha) = \sum_j d_1(j, r^*, M, \alpha) + \sum_j d_2(j, r^*, M, \alpha)$, $j \in q_1 \cup q_2$. Let us prove the admissibility of $h_2(M, \alpha)$ firstly. The 1-sgn$(M(r_i))$ in $h_2(M, \alpha)$ indicates that the heuristic function considers waiting time of partial jobs at the time $g(M, \alpha)$, while irrespective of the waiting time of all the jobs after the vertex (M, α). Thus, $\sum_i (1 - \mathrm{sgn}(M(r_i))) \cdot WT_i \cdot M({}^{\cdot}({}^{\cdot}r_i)) \leq \sum_j d_2(j, r^*, M, \alpha)$. Since the $X(t(r_i)^{\bullet})$ in $h_2(M, \alpha)$ hypothesizes that jobs are all processed on the most efficient machines, it suggests that $\sum_i \sum_m \sum_j x(j, r_{im}, g(M, \alpha)) \cdot (X(t(r_i)^{\cdot}) + RT(j, r_{im})) \leq \Sigma_j d_1(j, r^*, M, \alpha)$. Based on these two inequations, we can conclude that $h_2(M, \alpha) \leq h^*(M, \alpha)$, i.e., the $h_2(M, \alpha)$ is admissible. Since $\forall r_i \in P_R$, $WT_i \geq 0$ and $M({}^{\bullet}({}^{\bullet}r_i)) \geq 0$ at time $g(M, \alpha)$, hence $h_1(M, \alpha) \leq h_2(M, \alpha)$, and then $h_1(M, \alpha)$ is also admissible.

The third heuristic function satisfies $h_3(M, \alpha) = 0$ at any node (M, α)

of the system, and it is evidently admissible. Even though PN would reach the goal marking M_f under $h_3(M, \alpha)$, the algorithm randomly searches the next node at (M, α). Thus, A* algorithm with $h(M, \alpha) = 0$ can be regarded as reprocessing the cylinder blocks with a random scheduling scheme. Suppose h and h' are two heuristic functions for a specific problem, h' is more informed than h if they satisfy (1) both h and h' are admissible; and (2) $h' > h$ at any node (M, α) [39]. Therefore, the $h_2(M, \alpha)$ is the most informed one amongst these three heuristic functions.

To avoid state space explosion and select the most promising markings for further exploration, the dynamic window search (DWS) technique from [40] is adopted to constrain the search space of RG. In a search window, the depth of a vertex (M, α) is expressed as $depth(M, \alpha)$, i.e., the amounts of transitions have been fired, or length of α. We suppose the variables *top_depth* and *bottom_depth* refer the maximum and minimum depths of search window, and the distance between the *top_depth* and *bottom_depth* is a constant D. Let the $\chi_e(l)$ and $\chi_u(l)$ denote the quantities of explored and unexplored nodes in the l-depth. The parameter *max_node, max_size* and *max_top* refer the maximum number of nodes generated at each vertex (M, α), the maximum number of nodes can be explored in each depth, and the maximum number of nodes contained in the *top_depth*. Three rules [40] guide the search window to move deeper towards the goal marking.

Rule 1: if the $\chi_u(top\_depth) \geq max\_top$, then abandon all the nodes at the *bottom_depth*, and $top\_depth = top\_depth + 1$, $bottom\_depth = bottom\_depth + 1$;

Rule 2: if the $\chi_u(bottom\_depth) = 0$, then $top\_depth = top\_depth + 1$, $bottom\_depth = bottom\_depth + 1$.

Rule 3: for a new vertex (M, α) generated in l-depth (l-depth is in current window), if the $\chi_e(l) < max\_size$ or $f(M, \alpha) < \max\{f(M_i, \alpha_i) | (M_i, \alpha_i)$ are the vertexes in l-depth$\}$, then insert vertex (M, α) in l-depth, else reject (M, α).

Since the nodes might be discarded completely at *bottom_depth* as shown in *Rule 1*, this rule could lead to the loss of promising vertexes. To mitigate this negative effect, we modified the *Rule 1* into *Rule 1′* which deletes irrelevant nodes through the comparison with average estimated makespan. Additionally, *Rule 4* should be utilized as a complement if *Rule 1′* is applied in DWS. If a node in a depth deeper than *top_depth* possesses the least estimated *f* suggests that nodes of bottom depth deleted *Rule 4* are unpromising.

Rule 1′: if the $\chi_u(top\_depth) \geq max\_top$, for a vertex (M, α) at the *top_depth*, and its estimated makespan $f(M, \alpha)$, the average f of unexplored nodes at the *top_depth* is denoted as f_{av}, if $f(M, \alpha) < f_{av}$, then discard the vertex (M, α).

Rule 4: if the $depth(M, \alpha) > top\_depth$, then abandon all the nodes at the *bottom_depth*, and $top\_depth = top\_depth + 1$, $bottom\_depth = bottom\_depth + 1$;

The framework to perform the A* algorithm is performed as follows.

Step 1: Put M_0 into the OPEN list, OPEN = $\{M_0\}$, CLOSED = Ø;

Step 2: If the OPEN = Ø, then terminate the algorithm with failure;

Step 3: Apply *Rule 2*;

Step 4: Extract the first marking M in the OPEN list (M has the minimal f value and $depth(M) < top\_depth$) and apply *Rule 4* if *Rule 1′* is adopted in step 7.4, put this marking into the CLOSED list;

Step 5: If $M = M_f$, then find the optimal path (sequence of transitions) from the final marking M_f backwards to the initial marking M_0 based on the pointers, and terminate the algorithm;

Step 6: Figure out the enabled transitions t_e, rank these t_e with estimated makespan f after the firing of t_e, in ascending order, let $T_e(M)$ denotes the set of the top *max_node* transitions at the node (M, α);

Step 7: While $T_e(M)$ is not empty;

Step 7.1: remove $t_e \in T_e(M)$, determine the successive marking M' after the firing of t_e, set the pointer from M' to M, calculate the $f(M', \alpha')$ and $g(M', \alpha')$;

Step 7.2: if M' has been in the OPEN list, then direct the pointer along the direction with the smallest $g(M', \alpha')$. If M' has been in the CLOSED list, then ignore this M', go to step 7;

Step 7.3: if M' is neither in OPEN list nor the CLOSED list, then apply *Rule 3* to determine whether put it into the OPEN list;

Step 7.4: apply *Rule 1* (or *Rule 1′*);

Step 7.5: go to step 7;

Step 8: Reorder the markings in the OPEN list in ascending sequence according to their estimated makespan *f*;

Step 9: Go to step 2.

The scheduling strategy determined by the A* algorithm is aimed at the minimization of the makespan. With the scheduling scheme, the MC method is applied to statistically examine the total energy consumption and the makespan of cylinder block remanufacturing under the uncertainty of processing time.

5. Computational results

The A* algorithm was performed on the MATLAB platform and run on the personal computer with 4.0 GB RAM and 2.4 G CPU under Windows 10. Parameters setting in DWS technique can be denoted by DWS(*D, max_node, max_top, max_size*), and DWS(1, 2, 10, 10) is adopted in the present study. Since the processing time of each operation is presented by optimistic value, most plausible value, and pessimistic value, the makespan optimization is conducted based on corresponding three situations, which implies the range of makespan under uncertainty of processing time, as shown in Tables 2 and 3. It should be noted that results under h_3 are the average value of 20 times simulations.

It can be observed from Table 2 that the makespans of three situations simulated under h_1 and h_2 are identical, which indicates the effectiveness of both heuristic functions. However, the number of explored nodes under h_2 is slightly less than that under h_1. As the heuristic function h_3 searches the vertexes in deeper depth randomly, the optimization results are significantly greater than those obtained by h_1 and h_2. More specifically, the scheduling schemes generated by h_1 and h_2 would save approximately 1.3 hr in the most plausible situation compared with these by h_3. Additionally, to search for the final marking, h_3 would explore a greater number of nodes to find the final marking.

Instead of *Rule 1* in DWS, *Rule 1′* would remarkably reduce the number of explored nodes under all the three heuristic functions as displayed in Table 3. Similarly, the simulation results are also identical when applying h_1 and h_2. The makespans determined by A* algorithm incorporating the h_3 are also greater than optimization results under other two heuristic functions. Inspection of Tables 2 and 3 indicates that the range of the makespan is (9.0667, 10.6667) hr under the uncertainty of processing time.

To demonstrate the efficiency and effectiveness of this approach, we applied the traditional GA to solve this scheduling problem. For one single simulation, its computation time would be approximately 350 s, while it takes less than 2 s using the algorithm in this study with the same MATLAB platform. The makespan determined by GA is (9.37, 10.17, 11.20). Therefore, the simulation results in Table 2 and the computation speed significantly outperform the conventional GA.

Ideally, scheduling scheme integrating the optimistic, most plausible, and pessimistic situations is presented in one single Gantt chart, i.e., displayed in the form of triangular Gantt chart as in the work of [41,42]. Nonetheless, it is merely suitable for the Gantt chart of small-

Table 2
Simulation results under three heuristic forms (apply *Rule 1*).

Situations	Makespan (hr)			Number of explored nodes		
	h_1	h_2	h_3	h_1	h_2	h_3
Optimistic	9.0667	9.0667	9.9533	592	537	691
Most plausible	9.8667	9.8667	11.1937	487	487	668
Pessimistic	10.6667	10.6667	11.7217	524	517	675

Table 3
Simulation results under three heuristic forms (apply *Rule 1′*).

Situations	Makespan (hr)			Number of explored nodes		
	h_1	h_2	h_3	h_1	h_2	h_3
Optimistic	9.0667	9.0667	9.6533	139	112	176
Most plausible	9.8667	9.8667	11.0433	456	453	170
Pessimistic	10.6667	10.6667	11.3317	445	453	154

scale problems. To explicitly plot the sequences of reprocessing, we take the most plausible situation as an example to illustrate the remanufacturing of cylinder blocks. The makespan in this case is 9.8667 hr and corresponding sequence of transitions in optimal scheduling scheme is $t_{11}(r_{11})$, $t_{11}(r_{12})$, $t_{11}(r_{13})$, $t_{12}(r_2)$, $t_{21}(r_{11})$, $t_{21}(r_{12})$, $t_{21}(r_{13})$, $t_{13}(r_{31})$, $t_{12}(r_2)$, $t_{21}(r_{11})$, $t_{21}(r_{12})$, $t_{12}(r_2)$, $t_{13}(r_{32})$, $t_{21}(r_{13})$, $t_{22}(r_2)$, $t_{13}(r_{31})$, $t_{14}(r_{41})$, $t_{15}(r_5)$, $t_{22}(r_2)$, $t_{14}(r_{42})$, $t_{16}(r_6)$, $t_{15}(r_5)$, $t_{22}(r_2)$, $t_{17}(r_7)$, $t_{16}(r_6)$, $t_{15}(r_5)$, $t_{14}(r_{41})$, $t_{18}(r_8)$, $t_{17}(r_7)$, $t_{16}(r_6)$, $t_{15}(r_5)$, $t_{18}(r_8)$, $t_{22}(r_2)$, $t_{17}(r_7)$, $t_{16}(r_6)$, $t_{15}(r_5)$, $t_{18}(r_8)$, $t_{17}(r_7)$, $t_{16}(r_6)$, $t_{15}(r_5)$, $t_{19}(r_9)$, $t_{22}(r_2)$, $t_{18}(r_8)$, $t_{17}(r_7)$, $t_{16}(r_6)$, $t_{15}(r_5)$, $t_{18}(r_8)$, $t_{17}(r_7)$, $t_{16}(r_6)$, $t_{22}(r_2)$, $t_{15}(r_5)$, $t_{18}(r_8)$, $t_{17}(r_7)$, $t_{16}(r_6)$, $t_{19}(r_9)$, $t_{18}(r_8)$, $t_{15}(r_5)$, $t_{17}(r_7)$, $t_{18}(r_8)$, $t_{16}(r_6)$, $t_{17}(r_7)$, $t_{18}(r_8)$, $t_{19}(r_9)$. The r_{ij} in the parenthesis indicates the machine or resource for the firing of transition. As mentioned above, different from traditional PN, tokens in the present model are distinguishable and represent the machines, which would contribute to the simplification of PN model. Fig. 8 describes the firing sequence of transitions. For example, transition t_{11} has been fired three times at initial time 0 for the processing of 1st, 2nd, and 3rd job, and the firing time intervals are from time 0 to 1.28, from 0 to 1.33, and from 0 to 1.42, respectively. As evident in this figure, transitions t_{11}, t_{12}, t_{13}, and t_{14} are exclusively for the severely damaged cylinder blocks, while t_{21} and t_{22} are fired for the slightly damaged components. The transition t_{19} represents the remanufacturing cleaning process and three components are processed simultaneously.

Similar to the conventional Gantt chart, Fig. 9 explicitly displays the processing of jobs on each machine along with time. Two successive operations can be processed continuously irrespective of the intermediate transportations. The Gantt chart for transitions (Fig. 8) and machines (Fig. 9) are matched with each other. For example, after the firing of transition t_{11}, machine r_{11}, r_{12}, and r_{13} process the 1st job, 2nd, and 3rd job from time 0 to 1.28, from 0 to 1.33, and from 0 to 1.42, respectively. Bar of transition t_{11} in Fig. 8 can be regarded as the overlap of bars of r_{11}, r_{12}, and r_{13} in this example.

Incorporating the monitor and control of energy and environmental performance into the manufacturing system has been one of the ongoing efforts and also the prerequisites for promoting manufacturing sustainability. Real-time measurement of energy consumption can more clearly identify the potentials of performance improvement. The PN-based event-driven model in this study would simulate the dynamic performance of energy use as shown in Fig. 10. The energy consumption rate (ECR) profile (red line) of the cylinder block remanufacturing system shows a general rising tendency before the starting of all the machines even though the ECR fluctuates during the processing period. ECR profile of the system at any time point can be regarded as the sum of ECR of each machine at its own operation state. Due to the large rate power of cleaning equipment, the ECR has three remarkable crests, i.e., cleaning equipment has been started three times for a batch, as indicated in Fig. 10. The area under the red line denotes the total energy consumption to complete the reprocessing of all the used cylinder blocks. Accumulated energy use (blue line) shows the dynamic behavior of energy consumption in the system, and its slope equals the ECR. The dashed lines in this figure indicate the energy consumption rate and accumulated energy use profile of another batch of used components. The solid lines and dashed lines are completely not overlapped, which implies that the processing of each batch component is not periodical. Fig. 10 reveals that the utilization of energy efficient remanufacturing cleaning machines would contribute to the reduction of overall energy consumption.

To estimate the possibility of makespan and energy consumption for a batch cylinder blocks to be reprocessed under the uncertain processing time, the MC simulation is conducted based on the A* algorithm with h_1 and DWS(1, 1, 10, 10). This MC method repeated 500 random samplings, and numerical results regarding the makespan and energy consumption are presented by the histograms in Fig. 11. According to the Kolmogorov-Smirnov (K-S) test of the goodness of fit, the distribution of makespan satisfies Burr (4P). The average makespan is 9.861 hr, which is consistent with the results in Tables 2 and 3. When the heuristic function h_3 is applied in A* algorithm, the average makespan obtained by MC simulation is 10.67 hr, which implies that the scheduling strategy would save about 0.8 hr. Under the scheduling strategy determined by the A* algorithm, the total energy consumption is in line with Weibull (3P) distribution as also tested by K-S method, and 397.16 kWh electricity would be consumed for a batch on average.

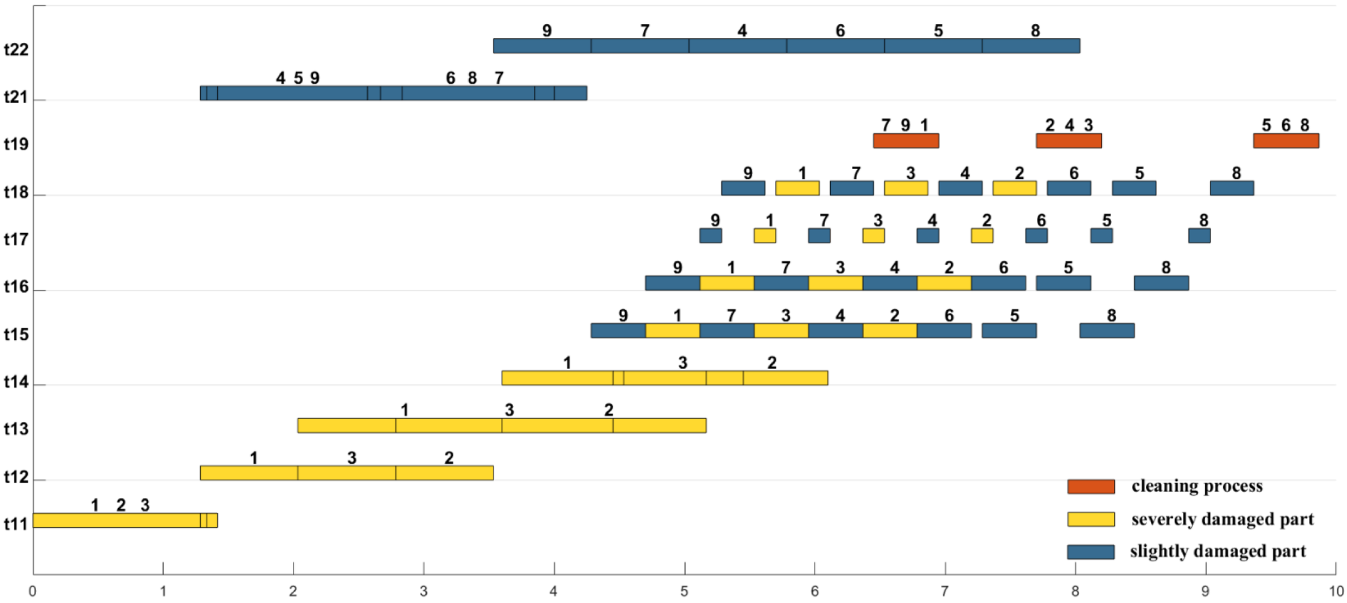


Fig. 8. Gantt chart for transitions.

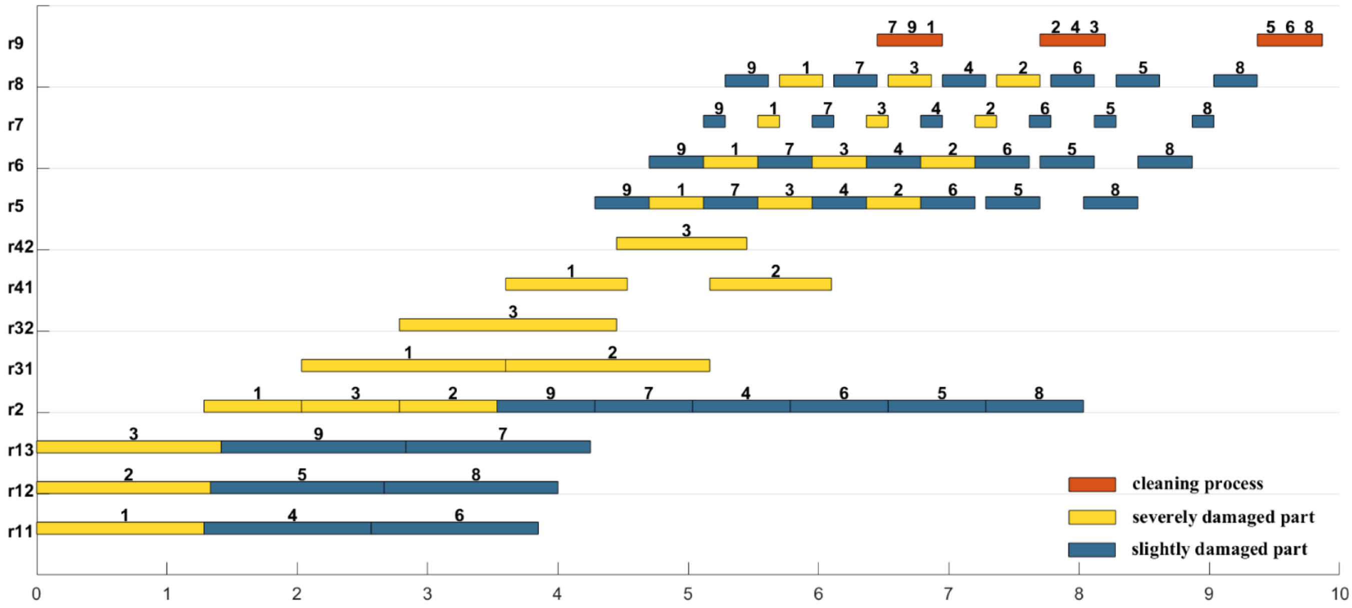


Fig. 9. Gantt chart for machines.

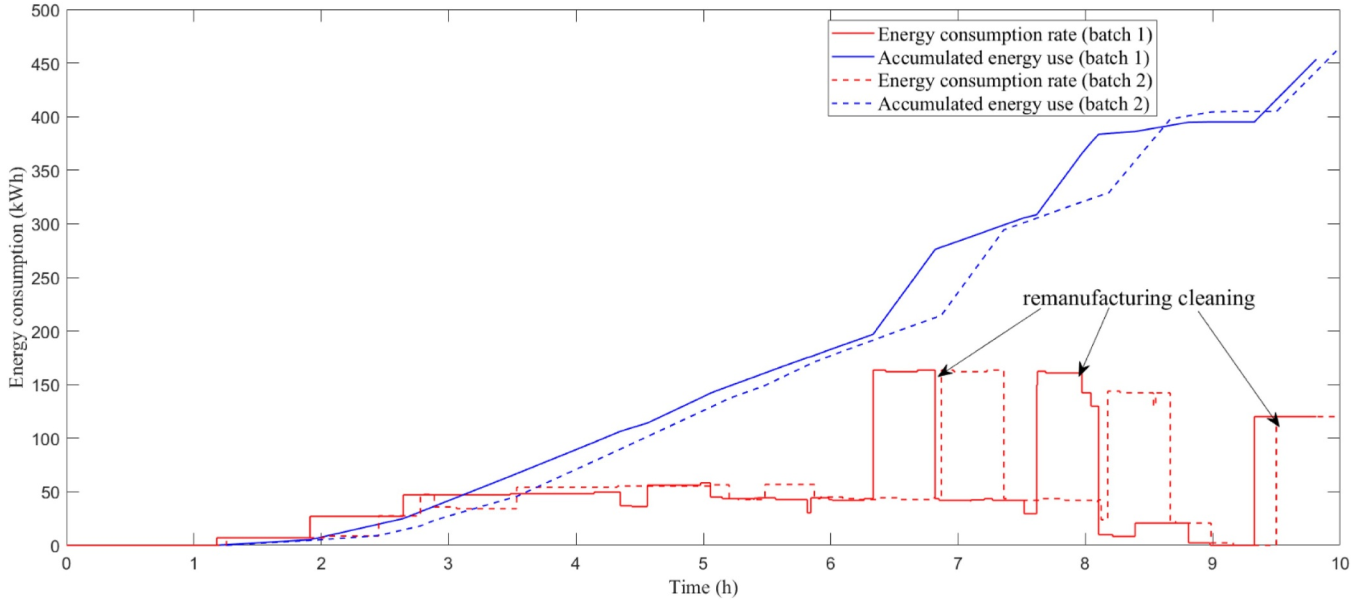


Fig. 10. Energy evolution behavior of different batches during their reprocessing periods.

The computation of energy consumption considered the processing operation and idleness of machines. For a specific batch, the energy saving primarily rests on the reduction of the idle period. Since damage degrees of used components are varying in different batches, a part with greater damage tend to cost longer time, and energy consumption is mainly determined by the processing time of operation on each machine.

6. Discussion and implication

As a mathematical formalized tool, PN is useful to design, model, and analyze discrete event systems [24]. It can explicitly describe the movement of jobs and selection of machines throughout the manufacturing system. Constraints are easily satisfied in the PN model such as processing one single or multiple components on a machine at a time, and a new operation of a job is initiated after the completion of prior operation. While the conventional scheduling model generally integrates these constraints into the program design stage, which would inevitably increase the difficulty of programming. Compared with other intelligent algorithms solving combinatorial optimization problems, A* search algorithm is more efficient, particularly in solving large-scale problems, and it would realize the trade-off between accuracy and efficiency of the solution by the control of DWS. In the present study, although the window in DWS is "small," the simulation also swiftly obtained a good solution. Therefore, the high efficiency of this algorithm makes it a candidate for real-time scheduling. In manufacturing practices, new jobs arrivals, machine breakdown, and order changes could frequently happen at any time. Therefore, scheduling decisions should be made in real-time. In these cases, the intelligent algorithms such as artificial bee colony algorithm and genetic algorithm are usually prohibitive due to the time-consuming computation particularly in the large-scale problems [43]. In this regard, one promising method is heuristic dispatching rules (HDR), for example, shortest processing time (SPT), first in and first out (FIFO), and longest processing time

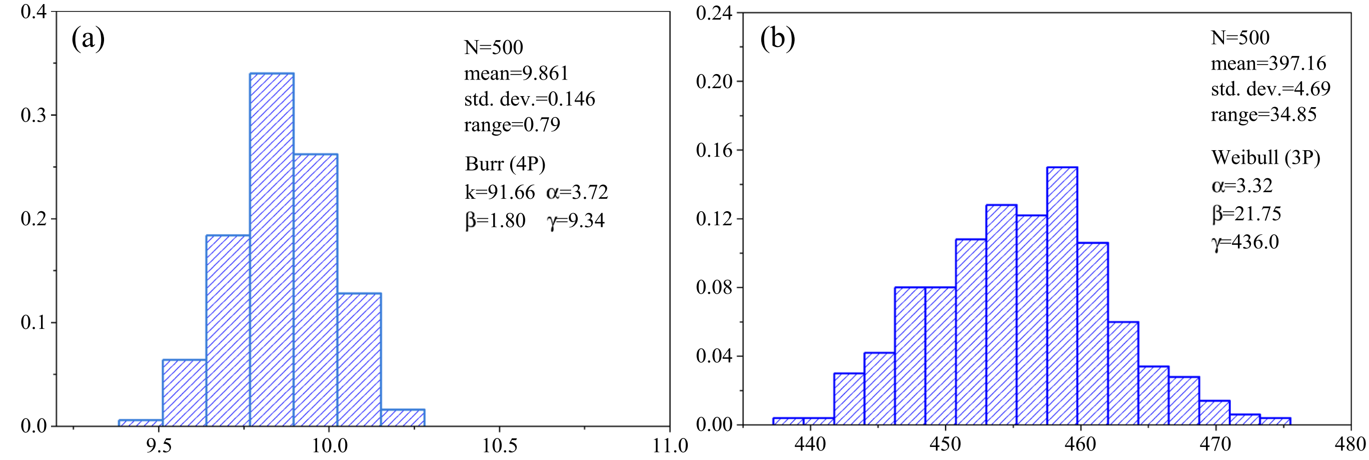


Fig. 11. Results of Monte Carlo simulation: (a) distribution of makespan (hr), (b) distribution of total energy consumption (kWh).

(LPT). The nodes exploration process in A* algorithm essentially belongs to HDR. For a specific scheduling problem, its computation efficiency is significantly greater than the intelligent algorithms. As also confirmed by Jiang [44], the applicability and adaptability of HDR are superior to the intelligent algorithms. Three heuristic functions were proposed for the A* algorithm. Even though h_2 was demonstrated to be the most informed heuristic function, it is not always superior in all cases compared with the other two heuristics. For example, the number of explored nodes by h_1 is less than that of h_2 as shown in Table 3. This phenomenon is consistent with the simulation of [24] which compared the results under three heuristics in A* algorithm. The selection of heuristics and setting of DWS parameters will affect the final results regarding makespan and the number of explored nodes. Counter-intuitively, the larger window, i.e., greater parameter setting would not necessarily generate better results. Following-up studies about the influence mechanism of the parameter setting on the final results are desirable.

Size of the present scheduling problem is 9 × 9 (number of jobs × number of operations). In this case, the statistical analysis of MC simulation results indicates that the optimal scheduling strategy would save approximately 0.8 hr. Since cylinder block is merely one of the engine components and the quantity of used parts considered in this study is limited, it is expected that the improvement of production efficiency would be greater if the scheduling strategy applied to the entire engine remanufacturing system. The total energy consumption under the scheduling strategy varies with different batches. MC simulation indicates that the range in 500 random batches is around 35 kWh. Total completion time and energy consumption in production scheduling are usually two conflict factors. In previous bi-objective scheduling studies [32,33,45] associated with the optimization of makespan and energy consumption, the source of energy use was confined in the machines processing components. Actually, the additional energy supporting the factory facilities such as air-conditioning and lighting are ignored in these studies. In this regard, the makespan of production is positively related to, rather than conflict with, the total energy use. Considering that these two objectives are highly coupled, the present study only concerned the makespan minimization irrespective of the optimization of energy consumption.

7. Conclusions

In the present study, we built a transition timed PN for the cylinder block remanufacturing. Distinguishable tokens in resource places are utilized to represent parallel machines, which would simplify and clarify the model without compromising the capability of system description. PN-based scheduling essentially transforms the problem into figuring out the optimal sequence of firing transitions. The possible sequences of firing transitions from initial marking to final marking create a tree structure. In this hierarchical graph, the optimal solution is a series of successive "branches" connecting the top "root" and the bottom. As an AI search algorithm, the heuristic A* algorithm was adopted to find the optimal sequence by graph traversal of the RG. We proposed three heuristic functions for this case and demonstrated their admissibility. To overcome the issue of state space explosion, a new rule-based DWS was integrated into this algorithm. The PN model could contribute to examining the evolution of energy use in dynamic remanufacturing processes. With the optimal scheduling strategy, the MC simulation was conducted to analyze the distributions of makespan and total energy consumption under uncertain processing time.

Simulation results performed with varying heuristics and dynamic window rules shown that range of the makespan is (9.0667, 10.6667) hr under optimal scheduling scheme. Makespans of optimistic, most plausible, and pessimistic situations determined under application of h_1 and h_2 are identical. Results also demonstrated the superiority of heuristic function h_2 and *Rule 1′* which tend to explore fewer nodes. The dynamic evolution of ECR indicated that the cleaning process is the dominated contributor to energy consumption and a potential "hotspot" for improvement of energy efficiency. MC simulation revealed that distributions of makespan and total energy consumption satisfied Burr (4P) and Weibull (3P), respectively, through the K-S test. The scheduling strategy obtained by A* algorithm would save approximately 0.8 hr on average for a batch used cylinder blocks.

This study only considered the uncertainty of processing time. However, the uncertainty in remanufacturing is multivariant and others might include stochastic return, quality and quantity variation, and uncertain remanufacturing rate. Thus, more efforts should be made to inclusively integrate these uncertainties into the scheduling of engine remanufacturing. Additionally, since the present study investigated only one component, a conceivable scheduling oriented the whole "seven pieces" of an engine could be remedied in the future. As the makespan and total energy consumption are highly coupled, the bi-objective scheduling regarding these two concerns and the exploration of their relationship are not captured here and remain topics for future research.

Acknowledgment

We highly appreciate the kind help from the manager, Huan Liu, at SINOTRUK, Jinan Fuqiang Power Corp. Ltd. Our gratitude is also extended to Natural Science Foundation of China (grant no. 51775086), Natural Science Foundation of China (grant no. 51605169), and Natural Science Foundation of Guangdong, China (grant no. 2014A030310345).

References

[1] L. Li, C. Li, Y. Tang, Y. Du, An integrated approach of reverse engineering aided remanufacturing process for worn components, Robot. Comput. Integr. Manuf. 48 (2017) 39–50, https://doi.org/10.1016/j.rcim.2017.02.004.

[2] J.H. Zhang, M. Chen, Assessing the impact of China's vehicle emission standards on diesel engine remanufacturing, J. Clean. Prod. 107 (2015) 177–184, https://doi.org/10.1016/j.jclepro.2015.03.103.

[3] Z. Jiang, Y. Jiang, Y. Wang, H. Zhang, H. Cao, G. Tian, A hybrid approach of rough set and case-based reasoning to remanufacturing process planning, J. Intell. Manuf. 30 (1) (2016) 19–32, https://doi.org/10.1007/s10845-016-1231-0.

[4] W. Xiang, C. Ming, Implementing extended producer responsibility: vehicle remanufacturing in China, J. Clean. Prod. 19 (6–7) (2011) 680–686, https://doi.org/10.1016/j.jclepro.2010.11.016.

[5] V.D.R. Guide, R. Srivastava, R.E. Kraus, Product structure complexity and scheduling of operations in recoverable manufacturing, Int. J. Prod. Res. 35 (11) (1997) 3179–3199.

[6] K.Z. Gao, P.N. Suganthan, Q.K. Pan, T.J. Chua, C.S. Chong, T.X. Cai, An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time, Expert Syst. Appl. 65 (2016) 52–67, https://doi.org/10.1016/j.eswa.2016.07.046.

[7] H. Wen, S. Hou, Z. Liu, Y. Liu, An optimization algorithm for integrated remanufacturing production planning and scheduling system, Chaos, Solitons Fractals 105 (2017) 69–76, https://doi.org/10.1016/j.chaos.2017.10.012.

[8] R. Zhang, S.K. Ong, A.Y.C. Nee, A simulation-based genetic algorithm approach for remanufacturing process planning and scheduling, Appl. Soft Comput. J. 37 (2015) 521–532, https://doi.org/10.1016/j.asoc.2015.08.051.

[9] I. Hatono, K. Yamagata, H. Tamura, Modeling and on-line scheduling of flexible manufacturing systems using stochastic petri nets, IEEE Trans. Softw. Eng. 17 (2) (1991) 126–132.

[10] Y. Lee, C. Chang, D.S. Wong, S. Jang, Petri-net based scheduling strategy for semiconductor manufacturing processes, Chem. Eng. Res. Des. 89 (3) (2010) 291–300, https://doi.org/10.1016/j.cherd.2010.07.005.

[11] S. Zhang, N. Wu, Z. Li, T. Qu, C. Li, Petri net-based approach to short-term scheduling of crude oil operations with less tank requirement, Inf. Sci. (NY). 417 (2017) 247–261, https://doi.org/10.1016/j.ins.2017.07.009.

[12] Ö. Basak, Y.E. Albayrak, Petri net based decision system modeling in real-time scheduling and control of flexible automotive manufacturing systems, Comput. Ind. Eng. 86 (2015) 116–126, https://doi.org/10.1016/j.cie.2014.09.024.

[13] H. Cao, H. Li, Simulation-based approach to modeling the carbon emissions dynamic characteristics of manufacturing system considering disturbances, J. Clean. Prod. 64 (2014) 572–580, https://doi.org/10.1016/j.jclepro.2013.10.002.

[14] Y. He, B. Liu, X. Zhang, H. Gao, X. Liu, A modeling method of task-oriented energy consumption for machining manufacturing system, J. Clean. Prod. 23 (1) (2012) 167–174, https://doi.org/10.1016/j.jclepro.2011.10.033.

[15] Q. Wang, F. Liu, C. Li, An integrated method for assessing the energy efficiency of machining workshop, J. Clean. Prod. 52 (2013) 122–133, https://doi.org/10.1016/j.jclepro.2013.03.020.

[16] K.E. Moore, A. Gungor, S.M. Gupta, A petri net approach to disassembly process planning, Comput. Ind. Eng. 35 (1–2) (1998) 165–168.

[17] S.M. Gupta, K.E. Moore, Petri net approach to disassembly process planning for products with complex AND/OR precedence relationships, Eur. J. Oper. Res. 135 (2) (2001) 428–449.

[18] E. Zussman, M. Zhou, Disassembly petri net approach to modeling and planning disassembly processes, Proc. 1998 IEEE Int. Symp. Electron. Environ, IEEE, Oak Brook, IL, USA, 1998, pp. 1–6.

[19] M.K. Tiwari, N. Sinha, S. Kumar, R. Rai, A Petri Net based approach to determine the disassembly strategy of a product, Int. J. Prod. Res. 7543 (5) (2010) 1113–1129, https://doi.org/10.1080/00207540110097176.

[20] C. Li, Research on Scheduling Algorithm Based on Petri Nets and Heuristic Search, Zhejiang University, China, 2015.

[21] T. Murata, Petri Nets: properties, analysis and applications, Proc. IEEE 77 (4) (1989) 541–580.

[22] B. Huang, Y. Sun, Y. Sun, C. Zhao, A hybrid heuristic search algorithm for scheduling FMS based on Petri net model, Int. J. Adv. Manuf. Technol. 48 (2010) 925–933, https://doi.org/10.1007/s00170-009-2329-8.

[23] W.M. Zuberek, Timed Petri nets definitions, properties, and applications, Microelectron. Reliab. 31 (4) (1991) 627–644.

[24] J. Luo, K. Xing, M. Zhou, Deadlock-free scheduling of automated manufacturing systems using Petri Nets and hybrid heuristic search, IEEE Trans. Syst. Man Cybern. Syst. 45 (3) (2015) 530–541.

[25] L. Han, K. Xing, X. Chen, F. Xiong, A Petri net-based particle swarm optimization approach for scheduling deadlock-prone flexible manufacturing systems, (2018) 1083–1096.

[26] K. Xing, M.C. Zhou, H. Liu, F. Tian, Optimal Petri-net-based polynomial-complexity deadlock-avoidance policies for automated manufacturing systems, IEEE Trans. Syst. Man Cybern. Part A Syst. Humans 39 (1) (2009) 188–199, https://doi.org/10.1109/TSMCA.2008.2007947.

[27] S. Kara, W. Li, Unit process energy consumption models for material removal processes, CIRP Ann. Manuf. Technol. 60 (1) (2011) 37–40, https://doi.org/10.1016/j.cirp.2011.03.018.

[28] W. Li, S. Kara, An empirical model for predicting energy consumption of manufacturing processes : a case of turning process, Proc. Inst. Mech. Eng. Part B J. Eng. Manuf. 225 (9) (2011) 1636–1646, https://doi.org/10.1177/2041297511398541.

[29] F. Le Bourhis, O. Kerbrat, J.-Y. Hascoet, P. Mognol, Sustainable manufacturing: evaluation and modeling of environmental impacts in additive manufacturing, Int. J. Adv. Manuf. Technol. 69 (9–12) (2013) 1927–1939, https://doi.org/10.1007/s00170-013-5151-2.

[30] H. Wang, Z. Jiang, Y. Wang, H. Zhang, Y. Wang, A two-stage optimization method for energy-saving flexible job-shop scheduling based on energy dynamic characterization, J. Clean. Prod. 188 (2018) 575–588, https://doi.org/10.1016/j.jclepro.2018.03.254.

[31] G.S. Liu, Y. Zhou, H.D. Yang, Minimizing energy consumption and tardiness penalty for fuzzy flow shop scheduling with state-dependent setup time, J. Clean. Prod. 147 (2017) 470–484, https://doi.org/10.1016/j.jclepro.2016.12.044.

[32] C. Lu, L. Gao, X. Li, J. Zheng, W. Gong, A multi-objective approach to welding shop scheduling for makespan, noise pollution and energy consumption, J. Clean. Prod. 196 (2018) 773–787, https://doi.org/10.1016/j.jclepro.2018.06.137.

[33] M. Dai, D. Tang, A. Giret, M.A. Salido, W.D. Li, Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm, Robot. Comput. Integr. Manuf. 29 (5) (2013) 418–429, https://doi.org/10.1016/j.rcim.2013.04.001.

[34] M. Dai, D. Tang, Y. Xu, W. Li, Energy-aware integrated process planning and scheduling for job shops, Proc. Inst. Mech. Eng. Part B J. Eng. Manuf. 229 (1_suppl) (2015) 13–26, https://doi.org/10.1177/0954405414553069.

[35] O.V. Arriaza, D.W. Kim, D.Y. Lee, M.A. Suhaimi, Trade-off analysis between machining time and energy consumption in impeller NC machining, Robot. Comput. Integr. Manuf. 43 (2017) 164–170, https://doi.org/10.1016/j.rcim.2015.09.014.

[36] G. Mejía, C. Montoya, Applications of resource assignment and scheduling with Petri Nets and heuristic search, Ann. Oper. Res. 181 (1) (2010) 795–812, https://doi.org/10.1007/s10479-010-0686-1.

[37] O.T. Baruwa, M.A. Piera, A. Guasch, Deadlock-free scheduling method for flexible manufacturing systems based on timed colored Petri Nets and anytime heuristic search, IEEE Trans. Syst. Man Cybern. Syst. 45 (5) (2015) 831–846.

[38] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, Reading, MA, USA, 1984.

[39] N.J. Nilsson, Artificial Intelligence: a New Synthesis, Morgan Kaufmann Publishers, Burlington, MA, USA, 1998.

[40] A.R. Moro, H. Yu, G. Kelleher, Advanced scheduling methodologies for flexible manufacturing systems using petri nets and heuristic search, Proc. 2000 IEEE Int. Conf. Robot. Autom, San Francisco, CA, USA, 2000, pp. 2398–2403.

[41] D. Lei, Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling, Appl. Soft Comput. J. 12 (8) (2012) 2237–2245, https://doi.org/10.1016/j.asoc.2012.03.025.

[42] S. Wang, L. Wang, Y. Xu, M. Liu, An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time, Int. J. Prod. Res. 51 (2013) 3778–3793, https://doi.org/10.1080/00207543.2013.765077.

[43] S. Li, Z. Wu, X. Pang, Job shop scheduling in real-time cases, Int. J. Adv. Manuf. Technol. 26 (7–8) (2005) 870–875, https://doi.org/10.1007/s00170-003-2051-x.

[44] Z. Jiang, Semiconductor Chip Manufacturing System Modeling and Optimal Scheduling Control, Shanghai Jiang Tong University Press, Shanghai, China, 2011.

[45] X. Yang, Z. Zeng, R. Wang, X. Sun, Bi-objective flexible job-shop scheduling problem considering energy consumption under stochastic processing times, PLoS One 11 (12) (2016) 1–13, https://doi.org/10.1371/journal.pone.0167427.