symmetry

Article

Petri-Net-Based Scheduling of Flexible Manufacturing Systems Using an Estimate Function

Gongdan Xu and Yufeng Chen *

Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau 999078, China; xgd222@163.com
* Correspondence: chyf01@163.com

Abstract: In this paper, a novel admissible estimate function is proposed to schedule flexible manufacturing systems (FMSs) by using heuristic search. The FMSs to be scheduled are modeled by P-timed Petri nets. The problem is to make the system evolve from the initial marking to a given final marking by firing a sequence of transitions. The structure of jobs in an FMS is always symmetrical to utilize the shared resources, but the processing time of each job is asymmetrical to reduce the global process time. By utilizing the structural symmetry of a Petri net model of an FMS, a partial reachability graph is generated such that the notorious state explosion problem is mitigated. For each generated marking, the proposed estimate function is used to provide an estimated cost for firing the transition sequence. Then, we can select the marking with the smallest cost from the generated markings and compute its successors. This process is continued until the system reaches the final marking. With the proposed method, the performance is evaluated in terms of the cost of the obtained transition firing sequence and the number of the expanded markings. The cost provided by the proposed estimate function is closer to the optimal cost than the previous work, i.e., the proposed method can find a transition firing sequence with less expanded markings and minimal process time from a marking to the final marking. Experimental results are used to demonstrate and evaluate the proposed approach.

Keywords: flexible manufacturing system (FMS); Petri net; scheduling; heuristic search

check for updates

Citation: Xu, G.; Chen, Y. Petri-Net-Based Scheduling of Flexible Manufacturing Systems Using an Estimate Function. *Symmetry* **2022**, *14*, 1052. https://doi.org/10.3390/sym14051052

Academic Editor: Juan Luis García Guirao

Received: 9 April 2022
Accepted: 18 May 2022
Published: 20 May 2022

1. Introduction

Flexible manufacturing systems (FMSs) are a classical type of discrete event system. They can produce multiple product types by sharing a limited number of resources [1–6]. These part types are concurrently processed by competing for these limited resources in a system, and each of them is manufactured by one of the predetermined routes. The production processes always have partially identical sequences to use the shared system resources. Hence, the structure of jobs in an FMS is always symmetrical in the sense of utilizing the shared resources, but the processing time of each job is asymmetrical. The goal of scheduling FMSs is to determine the assignment of resources in the process of production by considering some criteria (for example, the time cost and deadlock avoidance). With routing flexibility, it becomes an important issue to develop an efficient approach for scheduling FMSs to obtain a sequence of operations such that some performance criteria are optimized and relevant constraints are met. However, it is generally difficult to find an optimal solution using typical methods in a short time. Hence, a number of researchers have adopted artificial intelligence, and heuristics methods are applied to the scheduling problem. Specifically, in recent years, some researchers have proposed robust approaches that eliminate the necessity of optimizing such schedules [7–10].

Petri nets (PNs) [11] are a powerful tool for the modeling and analysis of FMSs, and for the scheduling of FMSs and various production systems [12–26]. They are suitable for representing the operations, resources and constraints of an FMS [15,27–34], as well as other behaviors in discrete event systems [35,36]. In particular, great attention has been paid to the scheduling problem of discrete event systems modeled with PNs and finite state automata in industrial and academic communities [12,13,16–19,22,24,37,38]. There are a number of groups working on the scheduling of FMSs by using P-timed PNs. In [39], Luo et al. present an anytime branch and bound scheduling problem of deadlock-prone FMSs modeled by P-timed PNs. By considering no-wait constraints, Wang et al. [40] develop a scheduling algorithm based on the P-timed PN model and heuristic search. The study in [41] focuses on solving the scheduling problem of minimizing the total energy consumption of FMSs. Le et al. [42] deal with uncertainties in the energy optimization of FMSs by using a weighted P-timed PN model. Baruwa et al. [43] touch upon the deadlock-free scheduling problem of FMSs by using timed colored PNs.

The A* algorithm, a well-known graph search algorithm, is used in a number of studies to deal with the scheduling problem of FMSs [14,23,44,45]. The A* strategy [46–48] is a well-known computer algorithm. It has been widely used in graph traversal and pathfinding in computer science. With a best-first search given the initial marking M_0, the A* strategy aims at finding the lowest-cost path from M_0 to the final marking M_f. The greatest advantage of these A* strategy-based methods is that they can obtain an optimal schedule if an admissible heuristic function is designed. However, the method is difficult to use for complex scheduling problems since an optimal solution cannot be found in a reasonable time. In [49], a dynamic weighting A* strategy is presented to reduce the computational cost. As stated in [49], the quality of the obtained solution is controllable since its cost is no more than $(1+\varepsilon)C^*$, where C^* denotes the optimal cost, i.e., the minimal processing time from M_0 to M_f. However, to achieve this, the depth of a solution should be estimated in advance. Furthermore, there is no guarantee of the optimality of the obtained solution. Huang et al. [50–53] have performed a great deal of work on this problem and proposed several heuristic search algorithms. Specifically, in [50], with an admissible heuristic function, they develop a weighted A* algorithm to schedule an FMS with alternative routes. By this method, it is not necessary to estimate the solution's depth in advance. More importantly, the cost of the obtained solution is controllable and it is no more than $(1+\varepsilon)C^*$.

Generally, an admissible estimate function for a marking is closer to the optimal time of all the paths passing it from the initial marking to the final marking, and the A* algorithm can more quickly find the optimal solution. The main contribution of the paper is to propose a more precise estimate function for a given marking. By using the A* algorithm, the proposed estimate function can find the optimal solution with fewer expanded markings than in the previous work. In this paper, motivated by the work in [50], by using a P-timed PN, we design a novel admissible heuristic function to schedule FMSs with alternative routes. Based on this function, a dynamic weighting scheduling strategy is presented. It uses only the current marking to obtain an estimate of the cost and eliminates the need for anticipating the depth of a solution. Furthermore, with the proposed method, the cost of a solution obtained by the proposed method is no more than $(1+\varepsilon)C^*$ ($\varepsilon \geq 0$, i.e., its quality is controllable). Finally, an FMS is used to demonstrate the method's applications and the improvement over the work in [50].

The remainder of the paper is organized as follows. Section 2 recalls the basics of P-timed PNs and FMSs. The novel heuristic function is proposed in Section 3. Section 4 presents the solution algorithm that uses the improved heuristic function. The experimental results for the proposed heuristic function are shown in Section 5. We conclude this paper in Section 6.

2. Preliminaries

Some basics of untimed PNs and P-timed PNs, and the modeling of FMSs, are recalled in this section. More details can be found in [11,50,54–57].

A (untimed) PN is a four-tuple $N = (P, T, F, W)$, where P and T are non-empty and finite sets of places and transitions, respectively, with $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ indicates a PN flow relation, represented by directed arcs from transitions to places or places to transitions. The weight of an arc is a mapping $W : (T \times P) \cup (P \times T) \to \mathbb{N}$, i.e.,

$W(x,y) > 0$ if $(x,y) \in F$; otherwise, $W(x,y) = 0$, where $\mathbb{N} = \{0, 1, 2, \ldots, \}$ denotes the set of nonnegative integers and $x, y \in P \cup T$.

A marking M of a PN N is a mapping: $P \rightarrow \mathbb{N}$. For a place p, $M(p)$ denotes the number of tokens in it at M. (N, M_0) is called a net system, where M_0 is an initial marking. Let $x \in P \cup T$ be a node of $N = (P, T, F, W)$. Then, the preset and postset of x are defined as ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ and $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$, respectively. We can extend the notation to a set of nodes $X \subseteq P \cup T$: ${}^\bullet X = \cup_{x \in X} {}^\bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$.

A transition $t \in T$ is enabled at a marking M if, for all $p \in {}^\bullet t$, $M(p) \geq W(p, t)$, which is denoted by $M[t\rangle$. Firing t at M, a new marking M' is generated such that, for all $p \in P$, $M'(p) = M(p) - W(p, t) + W(t, p)$, denoted by $M[t\rangle M'$. If there is a sequence of transitions $\sigma = t_0 t_1 \ldots t_n$, as well as markings $M_1, M_2, \ldots, M_n$, such that $M[t_0\rangle M_1 [t_1\rangle M_2 \ldots M_n [t_n\rangle M''$ holds, M'' is said to be reachable from marking M, denoted by $M[\sigma\rangle M''$. $R(N, M_0)$ is called the set of reachable markings of a net system (N, M_0), consisting of all the markings reachable from the initial marking M_0, i.e., $R(N, M_0) = \{M \in \mathbb{N}^{|P|} | \exists \sigma \in T^* : M_0[\sigma\rangle M\}$.

A P-timed PN is a six-tuple $N = \{P, T, I, O, M, D\}$, where:

- $P = \{p_1, p_2, \ldots, p_\mathrm{n}\}$, $n \in \mathbb{N}^+ = \{1, 2, \ldots\}$ is a set of places;
- $T = \{t_1, t_2, \ldots, t_\mathrm{m}\}$, $m \in \mathbb{N}^+$ is a set of transitions;
- $I : P \times T \rightarrow \mathbb{N} = \{0, 1, 2, \ldots\}$ is a mapping for defining directed arcs from P to T;
- $O : P \times T \rightarrow \mathbb{N}$ is a mapping for defining directed arcs from T to P;
- $M : P \rightarrow \mathbb{N}$ is an n-dimension vector with the i-th element $M(p_\mathrm{i})$ denoting the token count in p_i, with M_0 being the initial one;
- $D : P \rightarrow R^+ \cup \{0\}$ is a mapping for describing the time delay associated with the places, with R^+ being a set of positive real numbers. Formally, D is an n-dimension vector whose i-th component $D(p_\mathrm{i})$ is the time delay with p_i.

For PN models of FMSs, there are generally three types of places: the set of idle places P^0, the set of operation places P_R and the set of resource places P_A [54–57]. The maximal number of products to be processed concurrently for a product type is given by the tokens in the corresponding idle place at the initial marking. In this paper, an idle place is divided into two places: a source one and a sink one. The source place is the one without input transitions, while the sink place is the one without output transitions. A token in a source place represents a raw part ready for processing, while a token in a sink one indicates a completed part. Resources (machines and robots, for example) in an FMS are modeled by resource places. The tokens in a resource place at the initial marking represent the available resource units. The operation places indicate the operations to be performed for the parts in the production sequences and are initially unmarked. Based on P-timed PNs, each operation place is associated with a time delay, representing the time taken for processing the according operation. The token in an operation place is available when its time delay elapses. Note that some operation places have no time delay since the corresponding operation can be finished immediately. In other words, the tokens in these operation places are available at any time. The aim of scheduling FMSs is to find a transition firing sequence such that all tokens in the source places are moved to the sink places, with the optimization of the processing time.

A running example in [50] is considered to illustrate the modeling of P-timed PNs. It has three resources, R1, R2, and R3, to process four types of jobs, J1, J2, J3, and J4, and their production processes are shown in Figure 1. **I** and **O** are the input and output for each job. A path from **I** to **O** implies that the job can be performed in a sequential manner using the resources in the path, and the operating time for each operation is shown below the resources. Each job has a number of operations. For example, J4 has three operations, which use R3, R3, and R1 in sequence. Thus, J4 can be finished by sequentially using R3, R3, and R1, with the corresponding operating times being 99, 76, and 93, respectively. Some operations may be processed by alternative resources. For instance, the first operation of J1 can be processed by R3 or R2 alternatively, with the operating time being 69 and 75, respectively. The P-timed PN for this system is shown in Figure 2.

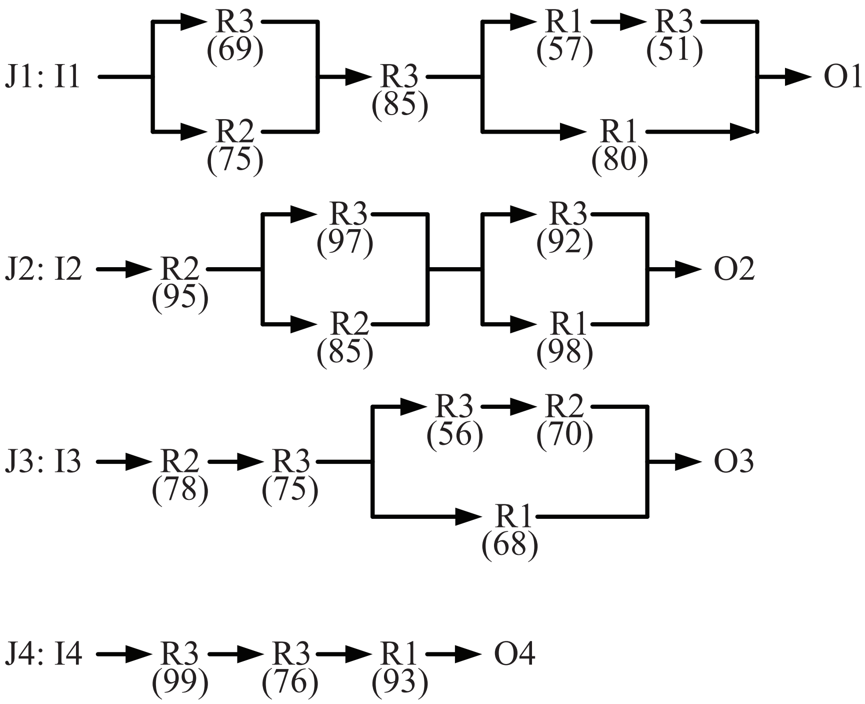

Figure 1. The four jobs.

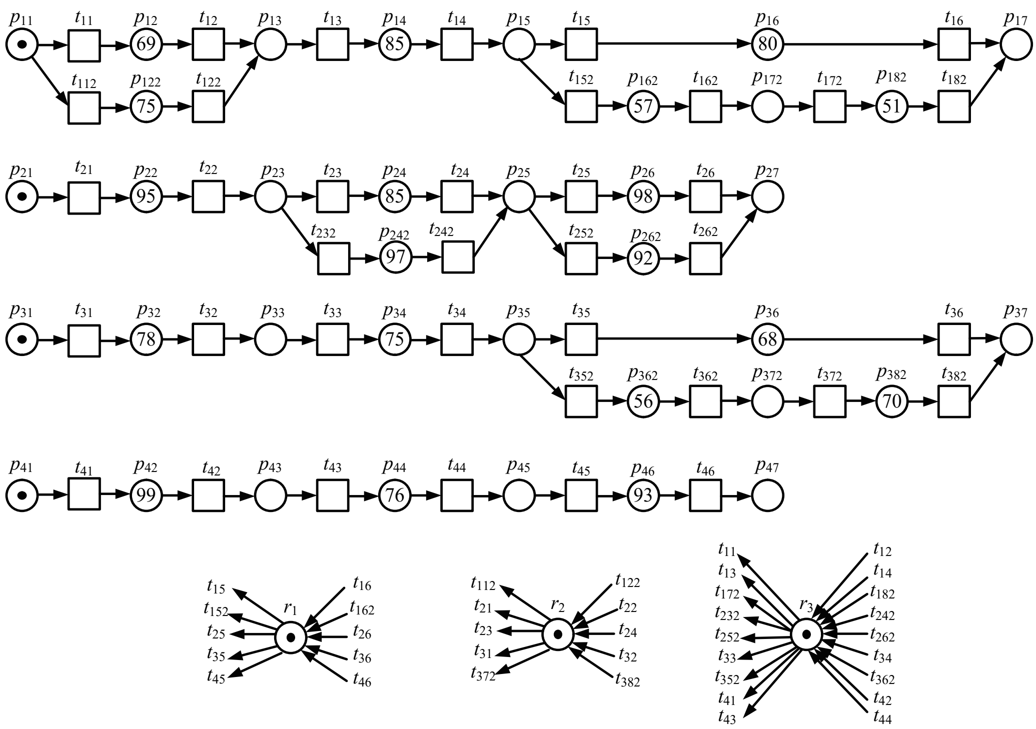

Figure 2. A P-timed PN model of an FMS.

In [58], with an A* algorithm and P-timed PNs, a heuristic search algorithm is developed. To perform this algorithm, the cost for a marking M is estimated by a function $f(M) = g(M) + h(M)$, which calculates the time required for the PN to evolve from M_0 to M_f via M along an optimal path. Specifically, $g(M)$ gives the current shortest time taken from M_0 to M, while $h(M)$ estimates the time needed from M to M_f. Function $h(M)$ is said to be admissible if it is no greater than all possible solutions obtained from any M [20,21], i.e.,

$$h(M) \leq h^*(M), \forall M \quad (1)$$

where $h^*(M)$ presents the shortest time needed from M to M_f. As stated in [45], the admissibility of $h(M)$ can guarantee the optimality of the obtained solution.

3. A Novel Heuristic Function

In [20,21,51,52], an admissible heuristic function is applied, as shown below:

$$h(M) = max_i\{\xi_i(M), i \in \{1, 2, \ldots, N_R\}\} \quad (2)$$

with $\xi_i(M)$ estimating the time needed for completing all the remaining operations that definitely need to be processed by resource r_i at M, and N_R is the number of resources. As stated in [50], the FMSs with alternative routes have some alternative routes for performing operations. In this case, the alternative routes of operations are not definitely applied. Hence, this heuristic function is inapplicable. In the following, a novel heuristic function is proposed by modifying (2) as:

$$h'(M) = max_i\{\xi'_i(M), i \in \{1, 2, \ldots, N_R\}\} \quad (3)$$

with $\xi'_i(M)$ estimating the minimal time needed for completing the remaining operations by resource r_i at marking M.

Next, we define this function in a formally mathematical way. With the PN model in Figure 2, we can explain the according definitions.

Let p_i be a place and p_e be the sink place of p_i. A sink place is always the sink place of a process. In Figure 2, p_{17} is the sink place of all operation places for Job 1. Similarly, p_{27}, p_{37}, and p_{47} indicate the sink places of all operation places for Job 2, Job 3, and Job 4, respectively. The problem in scheduling an FMS is to decide on a transition firing sequence that can transform all tokens in the initial marking to the sink places by minimizing the total processing time. For this example, the goal is to find a transition firing sequence that can transform the tokens in p_{11}, p_{21}, p_{31}, and p_{41} to p_{17}, p_{27}, p_{37}, and p_{47}, respectively, and requires minimal processing time.

A path from p_1 to p_k is defined as $\theta_{p_1 \to p_k} = \{p_1 p_2 \ldots p_k\}$, and $\exists t \in T$, s.t. $I(p_{i,t}) > 0$ and $O(p_{i+1,t}) > 0$, $i \in \{1, 2, \ldots, k-1\}$. Let p_i be a place and $\theta_{p_i \to p_e}$ a path from p_i to p_e, where p_e is the sink place of p_i. All paths from p_i to p_e are denoted as $\Theta_{p_i \to p_e}$. Function $\tau_j(\theta_{p_i \to p_e})$ denotes the remaining processing time when using r_j to make a token in p_i to reach p_e along the path $\theta_{p_i \to p_e}$.

Now, we introduce a least remaining work time vector φ_{r_j} for each resource r_j. Then, an element $\varphi_{r_j}(p_i)$ in vector φ_{r_j} is defined as

$$\varphi_{r_j}(p_i) = min\{\tau_j(\theta_{p_i \to p_e}) | \forall \theta_{p_i \to p_e} \in \Theta_{p_i \to p_e}\} \quad (4)$$

In (4), $\varphi_{r_j}(p_i)$ indicates the least remaining processing time of using r_j for marking a token in p_i to reach the sink place p_e. For the PN in Figure 2, it has four paths from p_{11} to p_{17}, i.e., $\Theta_{p_{11} \to p_{17}} = \{\theta_1, \theta_2, \theta_3, \theta_4\}$, where the four paths are shown in Table 1. The processing time for path θ_i ($i \in \{1, 2, 3, 4\}$) with r_j ($j \in \{1, 2, 3\}$ being used is denoted by $\tau_j(\theta_i)$, as shown in the last three columns. Then, we have $\varphi_{r_1}(p_{11}) = min\{\tau_1(\theta_i), i \in \{1, 2, 3, 4\}\} = 57$. Similarly, we can find all elements in φ_{r_1}, i.e.,

$$\varphi_{r_1} = [57\ 57\ 57\ 57\ 57\ 57\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 93\ 93\ 93\ 93\ 93\ 0\ 0\ 0\ 0\ 0] \quad (5)$$

where elements in φ_{r_1} and the corresponding places are shown in Table 2.

Table 1. The four paths in $\Theta_{p_{11} \to p_{17}}$ in Figure 2.

θ_i	Paths	$\tau_1(\theta_i)$	$\tau_2(\theta_i)$	$\tau_3(\theta_i)$
θ_1	$p_{11}p_{12}p_{13}p_{14}p_{15}p_{16}p_{17}$	80	0	69 + 85
θ_2	$p_{11}p_{122}p_{13}p_{14}p_{15}p_{16}p_{17}$	80	75	85
θ_3	$p_{11}p_{12}p_{13}p_{14}p_{15}p_{162}p_{172}p_{182}p_{17}$	57	0	69 + 85 + 51
θ_4	$p_{11}p_{122}p_{13}p_{14}p_{15}p_{162}p_{172}p_{182}p_{17}$	57	75	85 + 51

Table 2. The elements in φ_{r_1} for the PN in Figure 2.

Place p	p_{11}	p_{12}	p_{122}	p_{13}	p_{14}	p_{15}	p_{16}	p_{162}	p_{172}	p_{182}	p_{17}	p_{21}	p_{22}	p_{23}	p_{24}	p_{242}	p_{25}	p_{26}	p_{262}	p_{27}
$\varphi_{r_1}(p)$	57	57	57	57	57	57	0	0	0	0	0	0	0	0	0	0	0	0	0	0
place p	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{362}	p_{372}	p_{382}	p_{37}	p_{41}	p_{42}	p_{43}	p_{44}	p_{45}	p_{46}	p_{47}	r_1	r_2	r_3
$\varphi_{r_1}(p)$	0	0	0	0	0	0	0	0	0	0	93	93	93	93	93	0	0	0	0	0

By considering resources r_2 and r_3, we have:

$$\varphi_{r_2} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 95\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 78\ 0\ 0\ 0\ 0 \\ 70\ 0\ 70\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0] \tag{6}$$

$$\varphi_{r_3} = [85\ 85\ 85\ 85\ 0\ 0\ 51\ 0\ 51\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 75\ 75 \\ 75\ 0\ 0\ 0\ 0\ 0\ 0\ 175\ 76\ 76\ 0\ 0\ 0\ 0\ 0\ 0\ 0] \tag{7}$$

where the elements in φ_{r_2} and φ_{r_3} have the same correspondences with the places as shown in Table 2 for φ_{r_1}.

Let M be a reachable marking. The token in a marked place p_i ($M(p_i) > 0$) is available when its time delay has run out. $D_M(p_i)$ represents the remaining processing time for a token in p_i. If a place p_i is marked at M, $D_M(p_i) = d(p_i)$. If time τ elapses, it becomes $D_M(p_i) = d(p_i) - \tau$. Thus, D_M represents the remaining processing time vector of M. The total remaining time $D_{r_j}(M)$ of r_j in the marked places at M is defined as:

$$D_{r_j}(M) = \sum_{M(p_i)>0, p_i \in H(r_j)} D_M(p_i) \tag{8}$$

where $H(r_j) = \{p | p \in r_j^{\bullet\bullet}\}$ is r_j's holder.

Then, the shortest processing time for marking M by using r_j, denoted by $\psi_{r_j}(M)$, is defined as:

$$\psi_{r_j}(M) = \sum_{M(p_i)>0} \varphi_{r_j}(p_i) + D_{r_j}(M) \tag{9}$$

Finally, we obtain a function $h_R(M)$ to estimate the time needed from M to M_f:

$$h_R(M) = max\{\psi_{r_j}(M), j = 1, 2, \ldots, N_R\} \tag{10}$$

Theorem 1. *$h_R(M)$ is admissible.*

Proof. Since $\psi_{r_j}(M)$ denotes the shortest processing time of a marking M for using r_j, it is no greater than the required time to process all operations from M to M_f. Thus, the maximum of $\psi_{r_j}(M)$ ($j = 1, 2, \ldots, N_R$) is no more than the time of any optimal firing sequence from M to M_f. In other words, $h_R(M)$ is admissible. □

In [50], Huang et al. provide a heuristic function $h_{RWT}(M)$ for a given marking M. First, they present a remaining work time (RWT) vector, where RWT_i indicates the minimum time needed for moving a token in p_i to its sink place. For instance, the RWT vector for the PN model in Figure 2 is a 40×1 vector:

$$\begin{aligned} \text{RWT} = [&234\ 165\ 165\ 165\ 80\ 80\ 51\ 0\ 51\ 0\ 0\ 272\ 177\ 177 \\ &92\ 92\ 92\ 0\ 0\ 0\ 221\ 143\ 143\ 68\ 68\ 70\ 0\ 70\ 0\ 0\ 278 \\ &179\ 179\ 93\ 93\ 0\ 0\ 0\ 0\ 0] \end{aligned} \tag{11}$$

Next, the heuristic function in [50] is defined as follows:

Definition 1 ([50])**.** *$h_{\text{RWT}}(M) = max_{p_i \in SM}(RWT_i + Mr_i)$, where Mr_i is the ith element that presents the remaining processing time for the ith place in an $n \times 1$ vector, with n being the number of places and SM the set of marked places.*

Note that Mr_i in Definition 1 equals $D_M(p_i)$. We also have the following result for $h_{\text{RWT}}(M)$ in [50].

Theorem 2 ([50])**.** *$h_{\text{RWT}}(M)$ is admissible.*

The estimate function $h_{\text{RWT}}(M)$ in [50] is admissible. However, it is not close enough to the optimal cost $h^*(M)$. In this case, it may lead to a longer makespan and more expanded markings. This fact motivates us to provide an admissible estimate function to provide an estimated cost that is closer to the optimal one, aiming to find a solution with a shorter makespan and less expanded markings. By combining $h_{\text{RWT}}(M)$ and $\psi_{r_j}(M)$, we obtain an improved heuristic function:

$$h_{\max}(M) = max\{h_R(M), h_{\text{RWT}}(M)\} \tag{12}$$

Theorem 3. *$h_{max}(M)$ is admissible.*

Proof. The conclusion holds immediately from Theorems 1 and 2. □

For any M reachable from the initial marking M_0, $h_{\max}(M) \geq h_{\text{RWT}}(M)$ holds. In this sense, we can claim that $h_{\max}(M)$ is closer to the optimal value $h^*(M)$ than $h_{\text{RWT}}(M)$ in [50].

4. Algorithm for Scheduling FMSs

Next, a novel weighting A* algorithm is developed by using $h_{\max}(M)$. The estimated cost $f(M)$, i.e., the makespan from M_0 to M_f via M along an optimal path, is presented below.

$$f(M) = g(M) + h_{\max}(M) + \varepsilon \frac{h_{\max}(M)}{h_{\max}(M_0)} h_{\max}(M) \tag{13}$$

Note that (13) is obtained by replacing $h(m)$ in [59] with the new estimate function $h_{\max}(M)$. Next, a search algorithm for scheduling FMSs by using P-timed PNs and the proposed heuristic function is presented as follows.

As stated in [50], Algorithm 1 can obtain a path with its cost being no more than $(1+\varepsilon)C^*$ if $h(M)$ is admissible, with C^* being the lowest cost among all paths from M_0 to M_f.

Algorithm 1 Scheduling FMSs.

Input: A P-timed PN $N = \{P, T, I, O, M, D\}$ with M_0 and M_f being the initial and final markings, respectively, and ε being a factor.
Output: A firing sequence from M_0 to M_f.

1: $N_{\text{OPEN}} := 1$. $\mathcal{M}_{\text{OPEN}}(N_{\text{OPEN}}) := M_0$. /* $\mathcal{M}_{\text{OPEN}}$ is a set of markings whose successors are not generated, $\mathcal{M}_{\text{OPEN}}(N_{\text{OPEN}})$ represents the N_{OPEN}-th marking in $\mathcal{M}_{\text{OPEN}}$, and N_{OPEN} indicates the number of markings in $\mathcal{M}_{\text{OPEN}}$. */
2: $N_{\text{CLOSED}} := 0$. $\mathcal{M}_{\text{CLOSED}} := \emptyset$. /* $\mathcal{M}_{\text{CLOSED}}$ is a set of markings whose successors have been generated and N_{CLOSED} is the number of markings in $\mathcal{M}_{\text{CLOSED}}$. */
3: $N_{\text{CLOSED}} := N_{\text{CLOSED}} + 1$. $\mathcal{M}_{\text{CLOSED}}(N_{\text{CLOSED}}) := \mathcal{M}_{\text{OPEN}}(N_{\text{OPEN}})$. $M := \mathcal{M}_{\text{OPEN}}(N_{\text{OPEN}})$. $N_{\text{OPEN}} := N_{\text{OPEN}} - 1$. /* Remove the last marking M with the minimal f from $\mathcal{M}_{\text{OPEN}}$ */
4: **if** ($M = M_f$) **then**
5: Terminate the algorithm and output $\sigma(M)$.
6: **end if**
7: **for** (each enabled t at M) **do**
8: Fire t and generate a new marking M', and compute the firing sequence $\sigma(M')$ of M' by adding t in $\sigma(M)$. Compute $g(M')$.
9: **if** ($\exists\mathcal{M}_{\text{OPEN}}(j)$ is equal to M') **then**
10: **if** ($g(M') < g(\mathcal{M}_{\text{OPEN}}(j))$) **then**
11: $g(\mathcal{M}_{\text{OPEN}}(j)) := g(M')$.
12: $\sigma(\mathcal{M}_{\text{OPEN}}(j)) := \sigma(M')$.
13: $$f(\mathcal{M}_{\text{OPEN}}(j)) = f(M') = g(M') + h_{\max}(M') + \varepsilon\frac{h_{\max}(M')}{h_{\max}(M_0)}h_{\max}(M').$$
14: **end if**
15: **else**
16: **if** ($\exists\mathcal{M}_{\text{CLOSED}}(j)$ is equal to M') **then**
17: **if** ($g(M') < g(\mathcal{M}_{\text{CLOSED}}(j))$) **then**
18: $g(\mathcal{M}_{\text{CLOSED}}(j)) := g(M')$.
19: $N_{\text{OPEN}} := N_{\text{OPEN}} + 1$.
20: $\mathcal{M}_{\text{OPEN}}(N_{\text{OPEN}}) := M'$.
21: $\sigma(\mathcal{M}_{\text{OPEN}}(N_{\text{OPEN}})) := \sigma(M')$.
22: $f(\mathcal{M}_{\text{OPEN}}(N_{\text{OPEN}})) = f(M') = g(M') + h_{\max}(M') + \varepsilon\frac{h_{\max}(M')}{h_{\max}(M_0)}h_{\max}(M')$.
23: **end if**
24: **end if**
25: Reorder $\mathcal{M}_{\text{OPEN}}$ in a decreasing way with respect to the value of f.
26: **end if**
27: **end for**
28: Go to Step 3

Let $g^*(M')$ be the optimal cost among all the paths from M_0 to M. Then, due to $h_{\max}(M') \leq h^*(M')$ and $g(M') = g^*(M')$, we have

$$\begin{aligned} f(M') &= g(M') + h_{\max}(M') + \varepsilon\frac{h_{\max}(M')}{h_{\max}(M_0)}h_{\max}(M') \\ &\leq g^*(M') + h^*(M') + \varepsilon\frac{h_{\max}(M')}{h_{\max}(M_0)}h^*(M') \\ &\leq C^* + \varepsilon h^*(M') \\ &\leq (1+\varepsilon)C^* \end{aligned}$$

This means that the cost of the path obtained by Algorithm 1 is controllable, i.e., no more than $(1+\varepsilon)C^*$. This fact can also be seen in Lemma 2 in [59] and the work in [50]. In the next section, we show that the proposed heuristic function can lead to a smaller number of extended markings than the one required by the work in [50].

5. Experimental Results

In this section, we apply the proposed method to an example from [50] for demonstration. With this example, the method is coded by the C++ program. Then, experiments are performed, and the results are shown in Table 3. The Gantt charts for the results obtained by the proposed method are shown in Figures 3 and 4. For comparison and to show the difference between the experimental results of the proposed heuristic function and the one proposed in [50], we also develop a C++ program to implement the method proposed in [50], and the obtained results are shown in the third column. It should be noted that we cannot obtain the same results as in Table 2 in work [50], although we use the same heuristic function proposed in [50]. In the following, discussions about the existing work in [50] and the proposed one are based on the experimental results obtained by our developed C++ programs.

Table 3. The experimental results.

ε	The Results in [50]		Our Implementation of [50]		Proposed Method	
	Makespan	Expanded Markings	Makespan	Expanded Markings	Makespan	Expanded Markings
0.0	427	1442	427	1165	427	969
0.1	427	662	427	905	427	749
0.2	435	596	427	872	427	684
0.3	435	621	427	908	427	805
0.4	435	398	427	905	427	876
0.5	435	237	496	876	427	357
0.6	435	229	509	777	505	193
0.7	435	283	547	1010	505	120
0.8	496	224	547	646	505	81
0.9	496	226	547	595	505	81
1.0	496	236	547	572	505	81

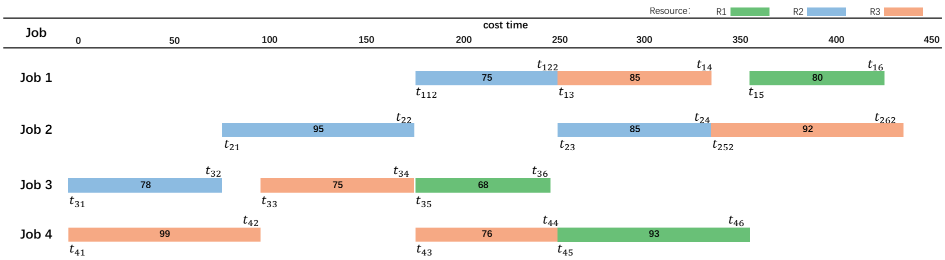

Figure 3. The Gantt chart for the proposed results with makespan = 427 in Table 3.

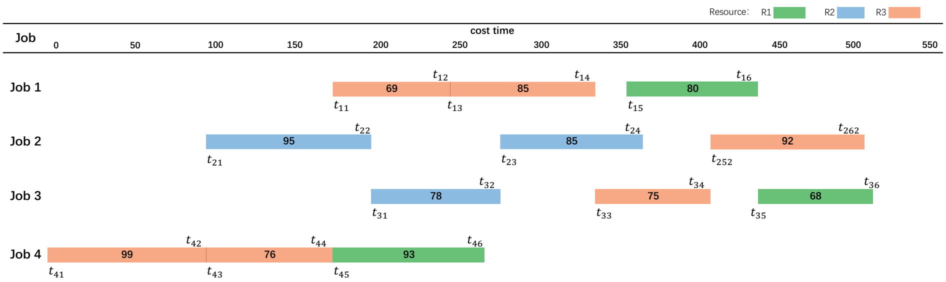

Figure 4. The Gantt chart for the proposed results with makespan = 505 in Table 3.

For this example, the optimized makespan and the number of markings that have been searched by using the heuristic function $h_{\mathrm{RWT}}(M)$ in [50] are 427 and 1165, respectively.

By using the proposed heuristic function $h_{\max}(M)$ in this work, these values are 427 and 969, respectively. This means that we can obtain the same makespan by searching less markings, i.e., $h_{\max}(M)$ proposed in this paper is better in terms of computational efficiency than $h_{\mathrm{RWT}}(M)$ in [50]. The second and third columns in Table 3 show that both the makespan and the number of extended markings obtained by this work are less than or equal to those obtained by our C++ program by using the method in [50]. From the table, the minimal number of extended markings to obtain an optimal makespan is 357 for the proposed method. In [50], the minimal number of extended markings to obtain an optimal makespan is 662. In summary, the proposed method outperforms the one in [50] in the sense of extended markings and the obtained makespan.

6. Conclusions

Motivated by the work in [50], this paper proposes a novel heuristic function for scheduling FMSs in the framework of P-timed PNs. The proposed heuristic function is admissible. A dynamic weighting heuristic scheduling strategy is applied by using the proposed function, aiming to find an optimal transition sequence such that the time taken for firing the sequence is optimized. By the proposed method, in the solution process, the estimation of a solution's depth in advance is not required. Furthermore, the obtained optimized cost is no greater than the optimal one by a factor $1 + \varepsilon$ ($\varepsilon \geq 0$). The performance of the proposed approach is verified via an example from the literature. With this example, results show that, by applying the proposed heuristic function, a solution with a shorter makespan can be obtained by searching less markings. In the future, we will focus on the improvement of the proposed approach by finding novel heuristic functions that are closer to the optimal cost.

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

Funding: This work was supported by the National Key R&D Project of China under Grant 2018YFB1700104 and the Science and Technology Development Fund, MSAR, under Grants 0064/2021/A2 and 0012/2019/A1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Thanks to the reviewers and editors for their constructive suggestions, which have been very useful for improving this paper.

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

References

1. Barkaoui, K.; Chaoui, A.; Zouari, B. Supervisory control of discrete event systems based on structure theory of Petri nets. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 2–5 October 1997; Volume 4, pp. 3750–3755.
2. Chen, Y.F.; Li, Z.W.; Khalgui, M.; Mosbahi, O. Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems. *IEEE Trans. Autom. Sci. Eng.* **2010**, *8*, 374–393. [CrossRef]
3. Chen, Y.F.; Li, Z.W. Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems. *Automatica* **2011**, *47*, 1028–1034. [CrossRef]
4. Chen, Y.F.; Li, Z.W.; Barkaoui, K.; Giua, A. On the enforcement of a class of nonlinear constraints on Petri nets. *Automatica* **2015**, *55*, 116–124. [CrossRef]
5. Fanti, M.P.; Zhou, M.C. Deadlock control methods in automated manufacturing systems. *IEEE Trans. Syst. Man Cybern. Part Syst. Humans* **2004**, *34*, 5–22. [CrossRef]
6. Huang, Y.S.; Jeng, M.; Xie, X.L.; Chung, D.H. Siphon-based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Syst. Man Cybern. Part Syst. Humans* **2006**, *36*, 1248–1256. [CrossRef]
7. Paprocka, I.; Skołud, B. Robust scheduling, a production scheduling model of failures. *Appl. Mech. Mater.* **2013**, *307*, 443–446. [CrossRef]

8. Módos, I.; Šůcha, P.; Hanzálek, Z. Algorithms for robust production scheduling with energy consumption limits. *Comput. Ind. Eng.* **2017**, *112*, 391–408. [CrossRef]
9. Sobaszek, L.; Gola, A.; Edward, K. Application of survival function in robust scheduling of production jobs. In Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, Prague, Czech Republic, 3–6 September 2011; Polish Information Processing Society: Prague, Czech Republic; pp. 575–578.
10. Sobaszek, Ł.; Gola, A.; Świć, A. The algorithms for robust scheduling of production jobs under machine failure and variable technological operation times. In *Innovations in Industrial Engineering*; Machado, J., Soares, F., Trojanowska, J., Ivanov, V., Eds.; Springer International Publishing: Cham, Switzwerland, 2022; pp. 56–67.
11. Murata, T. Petri nets: Properties, analysis and applications. *Proc. IEEE* **1989**, *77*, 541–580. [CrossRef]
12. Bai, L.P.; Wu, N.Q.; Li, Z.W.; Zhou, M.C. Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 1456–1467. [CrossRef]
13. Hou, Y.; Wu, N.Q.; Zhou, M.C.; Li, Z.W. Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *47*, 517–530. [CrossRef]
14. Li, C.; Wu, W.M.; Feng, Y.P.; Rong, G. Scheduling FMS problems with heuristic search function and transition-timed Petri nets. *J. Intell. Manuf.* **2015**, *26*, 933–944. [CrossRef]
15. Wu, N.Q.; Zhou, M.C.; Li, Z.W. Short-term scheduling of crude-oil operations: Enhancement of crude-oil operations scheduling using a Petri net-based control-theoretic approach. *IEEE Robot. Autom. Mag.* **2015**, *22*, 64–76. [CrossRef]
16. Wu, N.Q.; Zhou, M.C. Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation. *IEEE Trans. Autom. Sci. Eng.* **2011**, *9*, 203–209.
17. Wu, N.Q.; Zhou, M.C. Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets. *IEEE Trans. Autom. Sci. Eng.* **2012**, *9*, 446–454.
18. Wu, N.Q.; Chu, F.; Chu, C.B.; Zhou, M.C. Petri net modeling and cycle-time analysis of dual-arm cluster tools with wafer revisiting. *IEEE Trans. Syst. Man Cybern. Syst.* **2012**, *43*, 196–207. [CrossRef]
19. Wu, N.Q.; Li, Z.W.; Qu, T. Energy efficiency optimization in scheduling crude oil operations of refinery based on linear programming. *J. Clean. Prod.* **2017**, *166*, 49–57. [CrossRef]
20. Xiong, H.H.; Zhou, M.C.; Caudill, R.J. A hybrid heuristic search algorithm for scheduling flexible manufacturing systems. In Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 22–28 April 1996; Volume 3, pp. 2793–2797.
21. Xiong, H.H.; Zhou, M.C. Scheduling of semiconductor test facility via Petri nets and hybrid heuristic search. *IEEE Trans. Semicond. Manuf.* **1998**, *11*, 384–393. [CrossRef]
22. Yang, F.J.; Wu, N.Q.; Qiao, Y.; Zhou, M.C.; Li, Z.W. Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *47*, 502–516. [CrossRef]
23. Yim, S.J.; Lee, D.Y. Multiple objective scheduling for flexible manufacturing systems using Petri nets and heuristic search. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Information Intelligence and Systems, Beijing, China, 14–17 October 1996; Volume 4, pp. 2984–2989.
24. Zhang, S.W.; Wu, N.Q.; Li, Z.W.; Qu, T.; Li, C.D. Petri net-based approach to short-term scheduling of crude oil operations with less tank requirement. *Inf. Sci.* **2017**, *417*, 247–261. [CrossRef]
25. Qiao, Y.; Wu, N.Q.; Yang, F.J.; Zhou, M.C.; Zhu, Q.H. Wafer sojourn time fluctuation analysis of time-constrained dual-arm cluster tools with wafer revisiting and activity time variation. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *48*, 622–636. [CrossRef]
26. Zhu, Q.H.; Zhou, M.C.; Qiao, Y.; Wu, N.Q. Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *48*, 389–400. [CrossRef]
27. Basile, F.; Cordone, R.; Piroddi, L. A branch and bound approach for the design of decentralized supervisors in Petri net models. *Automatica* **2015**, *52*, 322–333. [CrossRef]
28. Chen, Y.F.; Li, Z.W.; Zhou, M.C. Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach. *IEEE Trans. Autom. Sci. Eng.* **2013**, *11*, 549–563. [CrossRef]
29. Chen, Y.F.; Li, Z.W.; Barkaoui, K.; Uzam, M. New Petri net structure and its application to optimal supervisory control: Interval inhibitor arcs. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *44*, 1384–1400. [CrossRef]
30. Chen, Y.F.; Li, Z.W.; Barkaoui, K.; Wu, N.Q.; Zhou, M.C. Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *47*, 364–379. [CrossRef]
31. Chen, Y.F.; Li, Z.W.; Al-Ahmari, A.; Wu, N.Q.; Qu, T. Deadlock recovery for flexible manufacturing systems modeled with Petri nets. *Inf. Sci.* **2017**, *381*, 290–303. [CrossRef]
32. Cong, X.Y.; Fanti, M.P.; Mangini, A.M.; Li, Z.W. Decentralized diagnosis by Petri nets and integer linear programming. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *48*, 1689–1700. [CrossRef]
33. Li, Z.W.; Hu, H.S.; Wang, A.R. Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets. *IEEE Trans. Syst. Man Cybern. Part Appl. Rev.* **2007**, *37*, 517–526. [CrossRef]
34. Uzam, M.; Li, Z.W.; Gelen, G.; Zakariyya, R.S. A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems. *J. Intell. Manuf.* **2016**, *27*, 1111–1129. [CrossRef]
35. Tong, Y.; Li, Z.W.; Giua, A. On the equivalence of observation structures for Petri net generators. *IEEE Trans. Autom. Control.* **2015**, *61*, 2448–2462. [CrossRef]

36. Tong, Y.; Li, Z.W.; Seatzu, C.; Giua, A. Verification of state-based opacity using Petri nets. *IEEE Trans. Autom. Control.* **2016**, *62*, 2823–2837. [CrossRef]
37. Wang, X.; Khemaissia, I.; Khalgui, M.; Li, Z.W.; Mosbahi, O.; Zhou, M.C. Dynamic low-power reconfiguration of real-time systems with periodic and probabilistic tasks. *IEEE Trans. Autom. Sci. Eng.* **2014**, *12*, 258–271. [CrossRef]
38. Wang, X.; Li, Z.W.; Wonham, W. Dynamic multiple-period reconfiguration of real-time scheduling based on timed DES supervisory control. *IEEE Trans. Ind. Inform.* **2015**, *12*, 101–111. [CrossRef]
39. Luo, J.; Zhou, M.; Wang, J.Q. AB amp;B: An Anytime Branch and Bound Algorithm for Scheduling of Deadlock-Prone Flexible Manufacturing Systems. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 2011–2021. [CrossRef]
40. Wang, X.; Xing, K.; Feng, Y.; Wu, Y. Scheduling of Flexible Manufacturing Systems Subject to No-Wait Constraints via Petri Nets and Heuristic Search. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 6122–6133. [CrossRef]
41. Li, X.; Xing, K.; Zhou, M.; Wang, X.; Wu, Y. Modified Dynamic Programming Algorithm for Optimization of Total Energy Consumption in Flexible Manufacturing Systems. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 691–705. [CrossRef]
42. Le, C.V.; Pang, C.K. Robust Total Energy Optimization of Flexible Manufacturing Systems Based on Renyi Mean-Entropy Criterion. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 355–367. [CrossRef]
43. Baruwa, O.T.; Piera, M.A.; Guasch, A. Deadlock-Free Scheduling Method for Flexible Manufacturing Systems Based on Timed Colored Petri Nets and Anytime Heuristic Search. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 831–846. [CrossRef]
44. Jeng, M.; Chen, S.C. A heuristic search approach using approximate solutions to Petri net state equations for scheduling flexible manufacturing systems. *Int. J. Flex. Manuf. Syst.* **1998**, *10*, 139–162. [CrossRef]
45. Lee, D.Y.; DiCesare, F. FMS scheduling using Petri nets and heuristic search. In Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, 8–13 May 1994; pp. 1057–1062.
46. Wikipedia. A* Search Algorithm. 2019. Available online: https://en.wikipedia.org/wiki/A*_search_algorithm (accessed on 8 April 2022)
47. Hart, P.; Nilsson, N.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
48. Abdulla, P.; Mayr, R. Computing optimal coverability costs in priced timed Petri nets. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science (LICS), Toronto, ON, Canada, 21–24 June 2011*; IEEE Computer Society: Washigton, DC, USA, 2011; pp. 399–408.
49. Pohl, I. The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computational issues in heuristic problem solving. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence, Stanford, CA, USA, 20–23 August 1973*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1973; pp. 12–17.
50. Huang, B.; Shi, X.X.; Xu, N. Scheduling FMS with alternative routings using Petri nets and near admissible heuristic search. *Int. J. Adv. Manuf. Technol.* **2012**, *63*, 1131–1136. [CrossRef]
51. Huang, B.; Sun, Y.M. Improved methods for scheduling flexible manufacturing systems based on Petri nets and heuristic search. *J. Control. Theory Appl.* **2005**, *3*, 139–144. [CrossRef]
52. Huang, B.; Sun, Y.; Sun, Y.M.; Zhao, C.X. A hybrid heuristic search algorithm for scheduling FMS based on Petri net model. *Int. J. Adv. Manuf. Technol.* **2010**, *48*, 925–933. [CrossRef]
53. Huang, B.; Zhou, M.; Abusorrah, A.; Sedraoui, K. Scheduling Robotic Cellular Manufacturing Systems With Timed Petri Net, A* Search, and Admissible Heuristic Function. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 243–250. [CrossRef]
54. Ezpeleta, J.; Colom, J.M.; Martinez, J. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Robot. Autom.* **1995**, *11*, 173–184. [CrossRef]
55. Park, J.; Reveliotis, S.A. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Trans. Autom. Control.* **2001**, *46*, 1572–1583. [CrossRef]
56. Tricas, F.; Martinez, J. An extension of the liveness theory for concurrent sequential processes competing for shared resources. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21st Century, Vancouver, BC, Canada, 22–25 October 1995; Volume 4, pp. 3035–3040.
57. Zhou, M.C.; DiCesare, F. Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources. *IEEE Trans. Robot. Autom.* **1991**, *7*, 515–527. [CrossRef]
58. Hsieh, F.S.; Chang, S.C. Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems. *IEEE Trans. Robot. Autom.* **1994**, *10*, 196–209. [CrossRef]
59. Pearl, J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*; Addison-Wesley Longman Publishing Co., Inc.: Reading, MA, USA, 1984.