A Dynamic Context-Aware Workflow Management Scheme for Cyber-Physical Systems Based on Multi-Agent System Architecture

Fu-Shiung Hsieh

Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan; fshsieh@cyut.edu.tw

Featured Application: This study provides a model driven approach in multi-agent systems to generate contextual information to guide workers/resources based on the Cyber World models in Cyber-Physical Systems.

Abstract: Although Cyber-Physical Systems (CPS) provides a paradigm to accommodate frequent changes in manufacturing sector, modeling and managing operations of CPS are challenging issues due to the complex interactions between entities in the system. Development of an effective context-aware workflow management system to guide the entities in the system is a critical factor to attain the potential benefits of CPS. In this paper, we will address the issue on the design of context-aware workflow management systems for CPS in IoT-enabled manufacturing environment. A CPS consists two parts, the Physical World and the Cyber World. To achieve the goal to design a context-aware information system for CPS, the Cyber World models of the entities in the system are constructed based on discrete timed Petri nets (DTPN) and a multi-agent system architecture in which each entity in the system is modeled as an agent to capture the interactions of entities in CPS. To develop context-aware workflow management systems for CPS, a Configuration/Scheduling Feasibility Problem and a Context Generation Problem in CPS are formulated. A condition for configuration/scheduling feasibility based on transformation of the Cyber World Models is established to develop an algorithm to generate contextual information to guide the operation of CPS. The proposed method is illustrated by examples. A series of experiments have been conducted to demonstrate the practicality of the proposed method in terms of computation time and response time. The results indicate that the computation time and total response time increase polynomially with respect to problem size parameters and show that the proposed method is effective in solving real problems.

Keywords: cyber-physical system; context-aware computing; multi-agent system; workflow

1. Introduction

Pressure due to frequent changes in demand from customers, global competition, environmental protection and global warming has posed new challenges in manufacturing sector. These challenges include how to achieve manufacturing agility to meet changing demands, how to develop flexible organization structure, and how to adopt emerging paradigms to face global competition by taking advantage of advanced information technology and manufacturing technology. Adoption of information technology in conjunction with advanced manufacturing technology and flexible organizational structure to achieve manufacturing agility and meet changing demands is a key issue for businesses to face global competition and attain sustainability. In the past years, several technologies and paradigms have been proposed to address the above issue. For example, Intelligent Manufacturing Systems (IMS) [1], Reconfigurable Manufacturing Systems (RMS) [2], Industry 4.0 [3,4], Cyber-Physical Systems (CPS), [5,6], and Internet of Things (IoT) [7]. Industry 4.0 aims to achieve smart factories characterized by adaptability, efficiency, reliability, safety,
usability and supporting dynamic supply chains [3,4] based on the increased integration of Information and Communication Technology (ICT) into production systems [5]. CPS combines mechatronics, ICT and computational decision-making components to control physical processes and systems to provide intelligence, responsiveness and adaptation [9]. With more and more things, sensors and devices getting connected, the world enters the era of Internet of Things (IoT) [10,11]. The IoT refers to the network of physical objects embedded with electronics, software, sensors and network connectivity which enables a rich set of interactions with the physical world characterized by machine-to-machine communications. It can be defined as a system of autonomous smart objects with sensing, information processing, and networking capabilities [12]. The IoT enables creation of innovative applications that may revolutionize the industry. Although the IoT provides the infrastructure for CPS to accommodate frequent changes in manufacturing sector, modeling and managing operations of CPS for manufacturing sector are challenging issues due to the complex interaction between entities in the system. Development of an effective context-aware information management system to guide the entities in the system is a critical factor to attain the potential benefits of CPS. This calls for the development of an effective decision support system to dynamically configure the system and generate the context-aware information for all relevant entities to guide their operations in CPS.

Context-aware computing is a paradigm that refers to the ability for an application to provide users with the information as needed and respond to users based on the context of use [13]. In manufacturing systems, an entity may be a machine, a worker or a resource. A context-aware system provides relevant information and/or services to the entities to improve service quality and operational efficiency based on “context”, which includes the information of identity, time, and activity [14]. Several survey papers on context-aware computing and the IoT in different application areas are available in [15,16]. The paper [17] surveys the design of implementation method for the components in context-aware ecosystems. There are several works on the development of context-aware information system for CPS [18]. For example, the paper [19] focuses on security solutions for context-aware CPS. The issue of programming context-aware CPS can be found in [20]. The work reported in [21] is dedicated to context-aware process guidance in CPS base on the use of augmented reality technique. The study of [22] concentrates on achieving context awareness for flexible manufacturing systems in CPS. Just like ergonomics [23], which focuses interactions between human and a system to optimize human well-being and performance, context-aware computing aims to enhance operational performance through timely provision of information. Different issues and studies in the literature may provide context-aware information relevant to manufacturing systems, e.g., inventory holding costs measurement [24] and scheduling [25]. However, generation of context-aware information for complex assembly processes such as the ones studied in [26-28] is a nontrivial research task. In addition, the issue to design context-aware workflow management systems for CPS with assembly processes is not addressed in the literature. To bridge the gap between the existing studies and the need for developing context-aware systems in CPS, we will address the design issue of dynamic context-aware workflow management systems for CPS. The goals of this papers are to develop a solution methodology to generate contextual information to guide workflows for a class of RMS with assembly operations modelled based on CPS and study the practicality of the proposed method in terms of computation time and response time through experiments.

The remainder of this paper is organized as follows. In Section 2, we briefly review the literature. In Section 3, we present a system architecture for RMS based on CPS paradigm. Section 4, we define the configuration/scheduling and context generation problems in CPS and briefly introduce the multi-level contract net protocol. We introduce the Cyber World Models of RMS in discrete timed Petri nets and a configuration/scheduling feasibility problem formulation in Section 5. A condition for configuration/scheduling feasibility based on transformation of Cyber World Models will be presented in Section 6. A context generation algorithm will be proposed in Section 7. Numerical examples will be presented
in Section 8. We will discuss the results and conclude this paper and introduce future research directions in Sections 9 and 10, respectively.

2. Literature Review

To develop an effective approach to achieving the goals of this study, we review relevant literature to select the methods and tools to develop a solution method. Due to the lack of a framework for design and validation of adaptive automation assembly systems in the Industry 4.0 era, Bortolini et al. propose a general framework to guide researchers, practitioners and companies to migrate from traditional manufacturing systems to advanced ones [4]. A self-adaptive smart assembly system (SASAS) has also been presented in [28] to address the design, engineering and testing of an adaptive assembly prototype. As the issue of generating context-aware information is not addressed in [4], the method proposed in this study complements the framework of [4]. The ability to cope with changing demand and technologies is more and more important in manufacturing sectors. RMS provides a paradigm to meet this requirements. In RMS, reconfigurability is a desirable property to adapt and allocate resources to meet order requirements. It has been studied in [29] for cellular manufacturing systems and [30] for CPS. CPS can be described as smart systems in which computational components in cyber space and physical components in physical space are seamlessly integrated with capabilities of communication and control to closely interact with each other to sense, monitor and control the changing state of the real world [9]. CPS involves a high degree of complexity with networked communication integrating computational and physical components to enable a new generation of “smart systems”. The concept of RMS and CPS also adopted in this study to develop a solution methodology for Industry 4.0.

There are many approaches to modelling, analysis and optimization of manufacturing systems, including optimization models [31,32], simulation models [33], Petri nets [34], and agent based models [35]. Each approach has strength and weakness. Although the optimization approach can optimize performance, it can only be applied to processes with fixed structure under specific assumptions. Furthermore, existing studies on optimization of RMS rely on mathematical programming cannot link to process modelling tools such as Petri nets. On the other hand, simulation approach is able to model a wide variety of production processes. However, it cannot be used to optimize the performance. The discrete event system approach such as Petri nets can be used to capture the properties of manufacturing systems. However, it is hard to optimize the performance based on discrete event system approach. Agent based approach allows modelling of flexibility, proactivity and negotiation capability of entities in the manufacturing systems. It is also hard to optimize the performance based on agent based approach. A pragmatic approach to address generation of contextual information in distributed architecture while optimizing the performance is to combine the advantages of existing approaches. Based on the above discussions, we propose an approach that adopts multi-agent systems as the underlying architecture, Petri net models as the internal models for agents and a transformation method that transforms the Petri net models into a graph based model suitable for applying the optimization methods.

A CPS for IoT-enabled reconfigurable manufacturing system consists two parts, the Physical World and the Cyber World. To achieve the goal to design context-aware information system for CPS, a multi-agent system model and a protocol based on the contract net [36] are adopted to capture the interactions and collaboration of entities in CPS. Each entity in the system is modeled as an agent. To develop the context-aware information system for CPS, the Cyber World models of the entities in the system are constructed. To construct the Cyber World models of the entities in the system, an easy to use and powerful modeling tool must be employed. Due to the superior capabilities in modeling and verification of workflows and processes in industrial systems, timed Petri nets (PNs) [37], which can be represented by the Petri Net Markup Language (PNML) [38] widely supported by software tools, are used in this study to construct the Cyber World
models of the entities in the system. However, the Cyber World models of the entities cannot work alone without a proper platform to facilitate the interactions and collaboration of the entities in the system. Therefore, a FIPA (http://www.fipa.org/, accessed on 25 January 2021) compliant multi-agent system infrastructure is used and a multi-level contract net protocol is applied to achieve task announcement, bidding, negotiation, and contract awarding. Based on the aforementioned Cyber World models, multi-agent system infrastructure and the multi-level contract net protocol, a context generation problem is formulated to generate the contextual information for the entities in the system to guide their operations. Although the Petri net is a powerful tool to model production activities in manufacturing systems, it is not an appropriate tool to solve the optimization problem to effectively generate the contextual information. A method is proposed to solve the context generation problem based on multi-agent systems (MAS) and the Cyber World models to develop an algorithm to generate contextual information to guide the operations of CPS. The proposed method will be illustrated by examples. In addition, to assess effectiveness of the proposed method, an important issue is to study how response time changes with respect to parameters of the problem. There are three parameters used in the solution algorithm, including time horizon, number of operations in a process and the demand. We perform a series of experiments to study the influence of these parameters on computation time of the solution method and the total response time. The results confirm applicability of the proposed method.

This study provides a model driven approach in multi-agent systems to generate contextual information to guide workers/resources based on the Cyber World models in Cyber-Physical Systems. One contribution of the approach is in the development of algorithms required for the generation of contextual information based on the Cyber World models specified by DTPN models. End users without the background of optimization theory only need to specify the DTPN models in DTPNs to apply the tool developed based on our approach. Another contribution of this study is to verify the computational efficiency of the proposed approach by a series of experiments. This paper is different from [39–41] which focus on scheduling and scalability issues in manufacturing system without considering context-awareness issue. It is also different from our previous works on context-aware workflow management reported in [42] which does not consider time factor. This paper is also different from [43] in that the interval based time model in [43] can only be used for temporal reasoning. This paper attempts to bridge the gap by introducing a proper time model to generate context-aware timing information for complex workflows in assembly systems.

3. A System Architecture for Reconfigurable Manufacturing System Based on CPS Paradigm

A reconfigurable manufacturing system described by a Cyber-Physical System in IoT-enabled environment consists of two parts, the Physical World and the Cyber World. Figure 1 shows the Physical World of RMS. It consists of the entities of workers, machines and parts in the system. The Cyber World of RMS is described by the models that represent resources such as workers, machines, parts, processes and data relevant to the decision making and operations of RMS. Figure 2 shows a Cyber World associated with Figure 1. In Figure 2, the entities of RMS in the Cyber World are modeled by agents in multi-agent system architecture. The models and data in the Cyber World provide an abstraction view of relevant entities in terms of the operational characteristics, dynamics, and architecture to achieve the production goals of RMS. Development of an appropriate Cyber World models for the entities in RMS is very important to effectively manage the Physical World of the CPS of RMS.
The Cyber World of RMS in Figure 2 consists of the several types of agents, including task agent, resource agents, process agents and context generation agents. All these agents interact with each other by applying some negotiation protocol provided by the MAS platform. Operations of RMS are driven by a goal, usually specified by an order. A task agent aims to achieve the goal of production according to the order received. A process agent denotes the production process, usually specified by a process (workflow) model. A process (workflow) model is used to describe the operations to be performed on a task and the precedence constraints of these operations. A resource agent represents a resource such as a machine or a robot in RMS. A resource model is used to describe the types of activities (operations) supported by a resource agent. The role of a context generation agent is to generate context-aware information to guide resources and workers in RMS.
optimization as well as analysis of solutions will be performed by the optimizer/analyzer to solve a configuration/scheduling feasibility problem. If the optimizer/analyzer can find a feasible solution, the solution will then be used as the input for the context generator. The context generator will create contextual information such as the operation lists for manufacturing resources to guide the operations of CPS. Although the architecture in Figure 2 provides a framework to design RMS based on the CPS paradigm, there still lack a study on how to systematically implement the model transformer, the optimizer/analyzer and the context generator for RMS. In this paper, we will focus on the development of models and a solution methodology needed to bridge the gap between theory and application in realizing the CPS paradigm for RMS.

![Figure 2. Providing Context-aware Information to Guide Resources/Workers.](image)

To generate context-aware information for resources/workers in RMS, a context generation agent must interact with relevant task agent, process agents, and resource agents. The internal architecture of a context generation agent include a model transformer module, an optimizer/analyzer module and a context generator. A context generation agent interacts with the relevant task agent, resource agents and process agents to obtain the goal, resource models, and process (workflow) models. The model transformer transforms the goal, resource models and process (workflow) models of agents into transformed models. Based on the transformed models obtained by the model transformer, optimization as well as analysis of solutions will be performed by the optimizer/analyzer to solve a configuration/scheduling feasibility problem. If the optimizer/analyzer can find a feasible solution, the solution will then be used as the input for the context generator. The context
generator will create contextual information such as the operation lists for manufacturing resources to guide the operations of CPS. Although the architecture in Figure 2 provides a framework to design RMS based on the CPS paradigm, there still lack a study on how to systematically implement the model transformer, the optimizer/analyzer and the context generator for RMS. In this paper, we will focus on the development of models and a solution methodology needed to bridge the gap between theory and application in realizing the CPS paradigm for RMS.

In the system architecture presented in Figure 2, there are several problems to be solved to implement a context-aware information system for RMS based on the CPS paradigm. These include collaborative interaction problem, model construction problem, model transformation problem, configuration/scheduling feasibility problem (CSFP) and context generation problem. In this section, these problems will be stated briefly. Each problem and relevant models or solution methods will be detailed in the rest of this paper.

The collaborative interaction problem aims to specify how agents in the system respond to requests issued by other agents. In this paper, a multi-level contract net protocol will be adopted. The details of the multi-level contract net protocol will be described in Section 4. The model construction problem is the problem to construct models for resource agents and process agents, including Cyber World process models and Cyber World resource models. In this paper, a discrete timed Petri nets (DTPN) [41] will be adopted to construct these models. The details for the construction of Cyber World process models and Cyber World resource models will be described in Section 5. The model transformation problem is the problem to devise a method to transform the original Cyber World process models and Cyber World resource models into a model that can facilitate optimization/analysis at later stage. The model transformation problem will rely on development of theory. Section 6 will be devoted to the development of a proper transformation method for the model transformation problem. The configuration/scheduling feasibility problem focuses on optimization/feasibility aspects of solutions/plans based on the transformed model. The problem will also be described in Sections 4–6. The context generation problem concentrates on generation of contextual information based on the solutions/plans found by the configuration/scheduling feasibility problem solver. The method to solve this problem will be detailed in Section 7.

4. Configuration/Scheduling Feasibility Problem and Context Generation Problem in CPS

In this section, we will state the configuration/scheduling feasibility problem and context generation problem in CPS. As the CPS consists of distributed resources, including machines, robots and workers interacting with each other autonomously based on the workflows to be performed, multi-agent systems (MAS) architecture is adopted to capture these characteristics and formulate these problems. To support interactions and negotiation of agents, it is assumed that a multi-level contract net protocol (CNP) based on [36] is used in the CPS. The multi-level CNP has specified the details about task announcement, bidding mechanism, contract awarding and negotiation. Therefore, we adopt the multi-level CNP in this study without re-inventing the wheel.

The proposed multi-agent system architecture consists of a set of resource agents and a set of process agents which work collaboratively to perform the operations specified by the task agent according to the orders received. To state the configuration/scheduling feasibility problem and the context generation problem in CPS, the notations in Table 1 are defined.
Table 1. Notations used in problem formulation.

| Symbol/Variable | Meaning |
|-----------------|---------|
| $N$             | the number of process agents |
| $WA$            | the set of process agents, where $WA = \{1, 2, \ldots, N\}$ |
| $n$             | a process agent, where $n \in WA$ |
| $R$             | the number of resource agents |
| $RA$            | the set of resource agents, where $RA = \{1, 2, \ldots, R\}$ |
| $a$             | a resource agent, where $a \in RA$ |
| $Q$             | the product demand of orders |
| $\psi$          | the due date of demand |
| $\Omega^k_n$   | the Cyber World model for the $k$-th operation of process agent $n$, where $n \in WA$ |
| $\Omega_n$     | the Cyber World model for process agent $n$, where $n \in WA$ |
| $\Omega = \{\Omega_n\}$ | the set of Cyber World models for all process agents in $WA$ |
| $A^k_{na}$      | the Cyber World model of the $k$-th operation of process agent $w_n$ performed by resource agent $a$, where $a \in RA$ |
| $A = \{A^k_{na}\}$ | the set of Cyber World models for all activities performed by resource agents in $WA$ |

We will defer the details of the notations above to later section to avoid distracting readers’ attention away from the problems to be addressed in this study. For example, construction of the Cyber World models $\Omega^k_n$ for all operations $k$ of process agent $n \in WA$ and the Cyber World models for all activities performed by resource agent $a \in RA$ will be introduced in later section. Let $\Omega = \{\Omega^k_n\}$ denote the set of Cyber World models for all process agents in $WA$. Let $A = \{A^k_{na}\}$ denote the set of Cyber World models for all activities performed by resource agents in $WA$. Based on the notations introduced above, suppose the Cyber World models $\Omega = \{\Omega^k_n\}$ for all operations of all process agents in $WA$ and the Cyber World models $A = \{A^k_{na}\}$ for all activities performed by all resource agents in $RA$ are given. The configuration/scheduling feasibility problem and the context generation problem are stated as follows, respectively:

Configuration/Scheduling Feasibility Problem (CSFP):

Given orders with product demand quantity $Q$ and due date $\psi$, a set of resource agents $RA$, a set of process agents $WA$ and the Cyber World models $\Omega = \{\Omega^k_n\}$ and $A = \{A^k_{na}\}$, determine whether there exist a set of process agents, $WA^*$, a set of resource agents, $RA^*$ and associated schedule $S_a$ for each agent $a \in RA^*$, to meet the product demand by the due date $\psi$, where $WA^* \subseteq WA$ and $RA^* \subseteq RA$.

The context generation problem to be studied in this paper is to generate context-ware information to guide resources/workers.

Context Generation Problem (CGP):

Generate contextual information for each agent $a \in RA^*$ based on the associated schedule $S_a$.

As no single agent can solve the above mentioned problem alone, collaboration among agents is necessary to solve the problems. The problems stated above rely on the development of solution algorithms based on interactions of agents in CPS in conjunction with application of optimization methods by individual agents to solve the problems. We will first present the interactions among agents in the problem solving processes and defer the details of the development of solution algorithms for the above problems until later sections.

Each workflow consists of one or more operations and there exists precedence constraints and among operations in each workflow. Moreover, there are precedence constraints among workflows as shown in Figure 3a. To perform an operation, a resource is needed. Therefore, there exists dependency among operations and resources as shown in Figure 3b.
To describe the problem solving processes, the following messages are defined:

Request for Proposal (RFP): RFP is a request sent by an agent at the downstream to its upstream agent. An RFP message is usually issued by either a task agent or a process agent. An RFP message can also be issued by a process agent to resource agents.

Propose (PRO): PRO is a response to an RFP message to indicate the proposal created by the agent sending the PRO message.

Award Contracts (AC): AC is a response to PRO. It indicates that the PRO message will be awarded.

Establish Contracts (EC): EC is a response to AC. It indicates that the AC message will be accepted.

Figure 4 shows the way a process agent handles an RFP message sent by a process agent at the downstream or a task agent. A process agent will wait for the arrival of an RFP message, which may be sent by either a task agent or a process agent at the downstream. It will first check whether the type of product demand is supported. If the type of product demand is supported, the procedure is terminated. Otherwise, the process agent proceeds to create an RFP message, sends it to the resource agents and waits for the PRO messages from the resource agents. Once the PRO messages are received, the process agent will determine whether a feasible solution exists by solving Problem 3 which is defined in Section 5. If there does not exist a feasible solution, the procedure will be terminated. Otherwise, the process agent proceeds to create an RFP message, sends it to upstream process agents and waits for the PRO messages from the upstream process agents. The RFP will carry a due date message obtained based on the feasible solution found by solving Problem 3 defined later in this paper. If PRO messages are received from the upstream process agents, the process agent will create a PRO message and send it to the process agent at the downstream or a task agent. If the task agent receives a PRO message from the upstream process agent and the PRO message indicates that the due date can be met, there exist a solution that can meet the deadline of the demand.
Dependency among workflows leads to complex interactions between the task agent and process agents in the solution finding processes. For the process agents in Figure 3a, interactions between the task agent and process agents is shown in Figure 5a. During the course of interactions, optimization algorithms will be invoked by the process agents. These optimization algorithms will be developed in later section based on the Cyber World Models and transformation of the Cyber World Models into graph models.

Similar to dependency among workflows, dependency among workflows and resources also results in complex interactions between process agents and resource agents in the problem solving processes. As W1 depends on resource agents R1 and R2, interactions among W1, R1 and R2 are shown in Figure 5b. As W2 depends on resource agent R3, interactions among W2 and R3 are shown in Figure 5c. As W3 depends on resource agents R4 and R5, interactions among W3, R4 and R5 are shown in Figure 5d.
Figure 5. (a) Interactions between the task agent and process agents. (b) Interactions between process agent W1 and resource agents R1 and R2. (c) Interactions between process agent W2 and resource agent R3. (d) Interactions between process agent W3 and resource agents R4 and R5.

Note that the development of methods/solution algorithms for the Configuration/Scheduling Feasibility Problem (CSFP) and the Context Generation Problem (CGP) for CPS requires the use of Cyber World models to facilitate the decision making processes. Construction of relevant Cyber World models for CPS will be introduced next.

5. Configuration/Scheduling Feasibility Problem Formulation Based on Cyber World Models of RMS in CPS

Construction of the Cyber World models $\Omega_n^k$ for all operations $k$ of all process agent $n \in WA$ and the Cyber World models for all activities performed by all resource agent $a \in RA$ will be introduced in this section. The Cyber World models are used to achieve the following objectives: (1) Specify the activities or operations to be performed by different types of agents, (2) formulate the configuration/scheduling feasibility problem and (3) generate and represent the contextual information provided to guide resources/workers in CPS. This section details the Cyber World models for the entities in CPS. The configuration/scheduling problem formulation and generation of contextual information for resources/workers will be presented later.
As there are different types of entities in CPS, we will propose the Cyber World models for each type of entities, including the Cyber World models for process agents and the Cyber World models for resource agents. To facilitate modeling and analysis, these Cyber World models are represented based on a class of Petri nets. As timing is an important characteristic that needs to be captured in the Cyber World models, we adopt discrete timed Petri nets (DTPN) \cite{41} to construct these models. A DTPN $G$ is described by a set of places, $P$, a set of transitions, $T$, a set of flow relation, $F \subseteq (P \times T) \cup (T \times P)$, an initial state or initial marking, $m_0$, and firing time for each transition specified by $\mu : T \rightarrow Z$, where $Z$ is the set of nonnegative integers. A DTPN $G$ can be described concisely by a five-tuple $G = (P, T, F, m_0, \mu)$ in which the initial marking $m_0 : P \rightarrow Z^{|P|}$. A marking $m \in Z^{|P|}$ of $G$ represents a state of the system indicating the number of tokens in each place. For transition $t$ in $T$, the set of input places of transition $t$ is represented by $^*t$ and the set of output places of $t$ is represented by $^t$. A transition $t$ enabled under a marking $m$ can be fired. Firing a transition $t$ once will remove one token from each place in $^*t$ and adds one token to each place in $^t$.

A bottom-up approach is adopted here to construct the Cyber World models. In constructing a Cyber World model with a DTPN, a place is used to represent a state in the workflow and a transition is used to represent an event that can bring the workflow from one state to another. To endow the workflow model with the capability to capture assembly operations, the workflow model proposed in this paper extends the one defined in \cite{41}.

**Definition 1.** The Cyber World model for the $k$-th operation of process agent $w_n$, $n \in WA$ is a DTPN $\Omega^k_n = \left( p^k_n, T^k_n, F^k_n, m^k_n, \mu^k_n \right)$ with $T^k_n = \left\{ t^k_n, t^*_{ne} \right\}$ and $P^k_n = P^k_n \cup \left\{ p^k_{ns} \right\} \cup \left\{ p^k_{no} \right\}$, where $t^k_n$ is the starting transition, $t^*_{ne}$ is the ending transition, $P^k_{ns} = \left\{ p^k_{ns} \right\}$ is the set of input places, $P^k_{no} = \left\{ p^k_{no} \right\}$ is the output place of the $k$-th operation and $t^k_n$ is a function specifying the firing time of each transition in $T^k_n$.

Figure 6a–e show the DTPN models $\Omega^1_1, \Omega^2_1, \Omega^1_2, \Omega^1_3$ and $\Omega^2_3$ for the operations $o_1, o_2, o_3, o_4$ and $o_5$ in Figure 3a. In Figure 6a–e, firing time is omitted for readability.

The Cyber World model is constructed by combining the Cyber World models of the operations involved. We define the following "\\" operator to combine two DTPNs.

**Definition 2.** Given two DTPNs, $G_1 = (P_1, T_1, F_1, m_{10}, \mu_1)$ and $G_2 = (P_2, T_2, F_2, m_{20}, \mu_2)$, $G_1\\G_2 = (P, T, F, W, m_0, \mu)$, where $P = P_1 \cup P_2$, $T = T_1 \cup T_2$, $F(p, t) = \begin{cases} F_1(p, t) & \text{if } p \in P_1 \text{ and } t \in T_1 \\ F_2(p, t) & \text{if } p \in P_2 \text{ and } t \in T_2 \end{cases}$, and $m_0(p) = \begin{cases} m_{10}(p) & \text{if } p \in P_1 \\ m_{20}(p) & \text{if } p \in P_2 \end{cases}$. 

Figure 6. (a) A DTPN model $\Omega^1_1$ for operation $o_1$; (b) A DTPN model $\Omega^2_1$ for operation $o_2$; (c) A DTPN model $\Omega^1_2$ for operation $o_3$; (d) A DTPN model $\Omega^1_3$ for operation $o_4$; (e) A DTPN model $\Omega^2_3$ for operation $o_5$. 

Appl. Sci. 2021, 11, 2030
Let $K_n$ denote the number of operations involved in the workflow of process agent $w_n$. The Cyber World model for process agent $w_n$ is defined as follows:

**Definition 3.** The Cyber World model for process agent $w_n$, where $n \in WA$ is an acyclic DTPN $\Omega_n = \{|k|_{\{1,2,...,K_n}\} \Omega_k|^n\}$.

Without loss of generality, it is assumed that only the input transition in the Cyber World model of the first operation may have multiple input places, i.e., $|p_{i,n}^k| \geq 1$ and $|p_{i,n}^k| = 1$ for $k \in \{2,\ldots,K_n\}$. If this condition is not satisfied, we can decompose the workflow of a process agent into several subworkflows with multiple input places allowed only for the first operation in each subworkflow and assign each subworkflow to a process agent to make the above condition satisfied. Figure 7a–c show the DTPN models, $\Omega_1$, $\Omega_2$ and $\Omega_3$ for process agents $w_1$, $w_2$ and $w_3$, where $\Omega_1 = \Omega_1^1||\Omega_2^2||\Omega_3^3$, $\Omega_2 = \Omega_1^1||\Omega_2^1$ and $\Omega_3 = \Omega_1^1||\Omega_3^1$.

![Figure 7](image)

**Figure 7.** (a) A DTPN model $\Omega_1$ for process agent $W1$; (b) A DTPN model $\Omega_2$ for process agent $W2$; (c) A DTPN model $\Omega_3$ for process agent $W3$.

Each operation of process agent $w_n$ must be performed by some resource agent. To describe the situation that $k$-th operation of process agent $w_n$ is performed by resource agent $a$, we define a DTPN $A_{na}^k = (p_{i,n}^k, t_{i,n}^k, F_{i,n}, m_{i,n}^k, \mu_{i,n}^k)$. In performing an operation, resource agent $a$ will start and end with the idle state place $r_a$. Therefore, $A_{na}^k = (p_{i,n}^k, t_{i,n}^k, F_{i,n}, m_{i,n}^k, \mu_{i,n}^k)$ is a cycle or circuit [37].

**Definition 4.** The Cyber World model of the $k$-th operation of process agent $w_n$ performed by resource agent $a$, where $a \in MA$, is represented by DTPN $A_{na}^k = (p_{i,n}^k, t_{i,n}^k, F_{i,n}, m_{i,n}^k, \mu_{i,n}^k)$. The initial marking $m_{i,n}^k(r_a)$ is the number of resources allocated, where $r_a$ is the idle state place. There is no common transition between $A_{na}^k$ and $A_{na}^{k'}$ for $k \neq k'$.

Figure 8a–e show the DTPN models, $A_{11}^1$, $A_{12}^2$, $A_{23}^3$, $A_{34}^4$ and $A_{35}^5$ for process agents $w_1$, $w_2$, and $w_3$. 
To model collaboration of resource agents and process agents, a collaboration model \( \Psi_n \) is obtained by combining the Cyber World model \( \Omega_n \) for process agent \( w_n \) and the Cyber World models of the activity performed by relevant resource agents. Let \( R^k \) denote the set of resource agents involved in performing the \( k \)-th operation in \( \Omega_n \). The collaboration model \( \Psi_n \) is defined as follows:

**Definition 5.** The Cyber World model for the collaboration of process agent \( w_n \) with relevant resource agents is a DTPN \( \Psi_n = (P_n, T_n, F_n, m_{n0}, \mu_n) = ||_{k \in \{1,2,...,K_n\}} (\Omega^k_n || A^k_{na}) \), where \( n \in WA \). The firing time \( \mu_n(t) \) for each transition \( t \in T_n \) is \( \mu^k_{na}(t) \).

Suppose \( R^1 = \{1\}, R^2 = \{2\}, R^3 = \{3\}, R^4 = \{4\}, R^5 = \{5\} \),

\[ \Psi_1 = (\Omega^1_1 || A^1_{11}) || (\Omega^1_2 || A^1_{12}) \],

\[ \Psi_2 = (\Omega^2_1 || A^2_{21}) || (\Omega^2_2 || A^2_{22}) \],

\[ \Psi_3 = (\Omega^3_1 || A^3_{34}) || (\Omega^3_2 || A^3_{35}) \].

Figure 9a–c show the DTPN models of \( \Psi_1, \Psi_2 \) and \( \Psi_3 \), respectively.

---

**Figure 8.** (a) A discrete timed Petri nets (DTPN) model for resource agent R1; (b) A DTPN model for resource agent R2; (c) A DTPN model for resource agent R3; (d) A DTPN model for resource agent R4; (e) A DTPN model for resource agent R5.

**Figure 9.** (a) A DTPN model \( \Psi_1 \) for collaboration of process agent W1, resource agent R1 and resource agent R2; (b) A DTPN model \( \Psi_2 \) for collaboration of process agent W2 and resource agent R3; (c) A DTPN model for resource agent R2; (c) A DTPN model \( \Psi_3 \) for collaboration of process agent W3, resource agent R4 and resource agent R5.
Due to the limited number of resources, $m^k_{ndkr}$ for performing the $k$-th operation of process agent $w_{kr}$, the firing time $\mu^k_{nkr} (t^k_{nr})$ for transition $t^k_{nr}$ the firing time $\mu^k_{nkr} (t^k_{nr})$ for transition $t^k_{nr}$ and the number of tokens that may enter the place $p^k_{nh}$ must satisfy capacity constraints and flow balance constraints. However, these constraints cannot be described in DTPN conveniently. As the above constraints must be satisfied, in the following section we will show that the evolution of a DTPN $\Psi_n = (P_n, T_n, F_n, m_{0n}, \mu_n)$ can be represented by a graph with constraints to facilitate the optimization processes and analysis of the system dynamics.

In this section, we will state the Configuration/Scheduling Feasibility Problem based on the DTPN models. Let $WA^*$ denote a set of process agents. The collaboration of the set of process agents in $WA^*$ and relevant resource agents $RA^* = \bigcup_{n \in WA^*} \bigcup_{k \in \{1,2,\ldots,K_0\}} R^k_n$ can be modeled by DTPN $\Psi = \|_{n \in WA^*} \Psi_n = (P, T, F, m_0, \mu)$. For the operations in three process agents and dependency among workflows and resources in Figure 3, based on the interactions between the task agent, process agents and resource agents in Figure 5a–d, collaboration of process agent $W_1$, resource agent $R_1$, resource agent $R_2$, process agent $W_2$, resource agent $R_3$, process agent $W_3$, resource agent $R_4$ and resource agent $R_5$ can be represented by $\Psi_1 \| \Psi_2 \| \Psi_3$ as shown in Figure 10 (all resources are assumed to stay in the corresponding idle state places initially and all parts are in the corresponding input places of process agent $W_1$ and process agent $W_2$).

![Figure 10. A DTPN model $\Psi_1 \| \Psi_2 \| \Psi_3$ for collaboration of process agent $W_1$, resource agent $R_1$, resource agent $R_2$, process agent $W_2$, resource agent $R_3$, process agent $W_3$, resource agent $R_4$, and resource agent $R_5$.](image)

Note that the CSFP aims to determine whether there exist a set of a set of process agents, $WA^*$, a set of resource agents, $RA^*$ and associated workflow schedule $S_\delta$ for each agent in $RA^*$ to perform operations in $WA^*$ such that demand quantity $Q$ can be met by
the deadline \( \theta \). Let \( f \) be the last process agent that produces \( Q \) final products for the task agent. A formal statement of this problem based on the DTPN model is as follows:

Centralized Configuration/Scheduling Feasibility Problem Formulation based on DTPN Model:

Given the requested product quantity \( Q \) and deadline \( \theta \), find a set \( WA^* \) of process agents and a set \( RA^* \) of resource agents such that there exists a firing sequence \( S \) of the DTPN \( \Psi \) that can bring the system to a marking \( m \left( \left( p^f_i \right) \right) = Q \) at \( t \) with \( t \leq \theta \), where \( S \) consists of the firing sequence \( S_a \) for each agent \( a \) in \( RA^* \) to perform operations specified by relevant process agent in \( WA^* \).

Although collaboration of agents can be modelled by Figure 10, the above centralized model and problem formulation cannot be used in distributed agent based manufacturing environment. Therefore, the Configuration/Scheduling Feasibility Problem (CSFP) will be formulated based on distributed computing architecture to take advantage of divide-and-conquer strategy.

We will propose a method to find a set \( WA^* \) of process agents, a set \( RA^* \) of resource agents and the firing sequence \( S_a \) for each agent \( a \) in \( RA^* \) in distributed multi-agent system architecture by applying the multi-level contract net protocol based on transformation of Cyber World models. The multi-level contract net protocol will be initiated by a task agent that represents an order. The task agent will send an RFP message to potential process agents that can produce the requested type of products. A potential process agent will send an RFP message to potential resource agents to acquire the required resources. Based on the potential process agent and the resource agents acquired, a DTPN model, \( \Psi_n \), will be constructed. It is necessary to determine whether there exists a firing sequence \( S_n \) for \( \Psi_n \) such that the final transition \( t_{nf} \) of \( \Psi_n \) is the last firing sequence in \( S_n \) and can be fired by the deadline. If there exists a firing sequence \( S_n \) for \( \Psi_n \) to fire \( t_{nf} \) by the deadline, the process agent will send an RFP message to potential process agents at its upstream. The processes repeats until a solution is found.

To support the above solution processes, we formulate a subproblem for each process agent. Each process agent will attempt to solve a subproblem based on collaboration with other resource agents and process agents according to the dependency on other processes and resources. We will study the condition for the existence of a solution based on solutions of subproblems. A problem formulation for the Configuration/Scheduling Feasibility Problem (CSFP) of each process agent based on the Cyber World model, DTPN \( \Psi_n = \left( P_n, T_n, F_n, m_n(0), \mu_n \right) = \| k \in \{1,2,...,K_n \} \left( \Omega^k_n \| a \in RA^k_n \right) \) is stated as follows:

**Problem 1.** Configuration/Scheduling Feasibility Problem Formulation for Individual Process Agent based on DTPN Model: Given the requested product quantity \( Q \), the deadline \( \theta \) set by the downstream product agent and a DTPN model \( \Psi_n \), determine whether there exists a set of resource agent, \( RA^*_n \), a firing sequence \( S_n \) for \( \Psi_n \) such that the final transition \( t_{nf} \) of \( \Psi_n \) is the last firing sequence in \( S_n \) and is fired \( Q \) times by the deadline.

6. A Condition for Configuration/Scheduling Feasibility Based on Transformation of Cyber World Models

Although the DTPN based Cyber World models defined above can capture the behaviors of agents, activities and resources, it is difficult to develop a solution algorithm to solve Problem 1 based on DTPN models. Therefore, we will transform a DTPN to a graph model and then find a solution based on the graph model. In this section, we first propose an algorithm to transform a DTPN to a graph model and study the properties that establish the connection between the DTPN and the graph model. Based on the properties, the problem to determine whether there exists a firing sequence \( s_n \) of \( \Psi_n \) and firing \( s_n \) will bring \( \Psi_n \) to a desired marking by a deadline can be formulated as an optimization problem defined on the graph model. Finally, we establish a configuration/scheduling feasibility condition based on the solution of the optimization problem.
Suppose the time horizon is \( T \). The dynamics of a DTPN Cyber World model \( \Psi_n = (P_n, T_n, E_n, m_{n0}, n_{n0}) \) of RMS can be described by the marking \( m_{nt} \) at different time period \( t \), where \( t \in \{1, 2, \ldots, T\} \). We first illustrate how to represent marking \( m_{nt} \) at different time period \( t \) by a graph \( DG_n(V_n, E_n) \). An algorithm to transform a DTPN \( \Psi_n \) to a graph \( DG_n(V_n, E_n) \) is proposed in Algorithms 1 as follows:

### Algorithms 1 DTPN to Graph Transformation (DTPN2GT) Algorithm

**Input:** \( \Omega_n = (P_n, T_n, E_n, m_{n0}, n_{n0}) \), \( \Phi \), where \( \Phi \) is greater than 0

**Output:** \( DG_n(V_n, E_n) \)

**Step 1:** Construct a source node \( S \) and a sink node \( U \)

**Step 2:** Construct a \((K_n + 1) \times T \) matrix of nodes, numbered from 1 to \((K_n + 1) \times T \).

**Step 3:** For \( k \in \{1, 2, \ldots, K_n\} \)

For \( t \in \{1, 2, \ldots, T - 1\} \)
- Add an arc connecting node \( t + (k - 1)T \) to node \( t + (k - 1)T + 1 \)
- Assign arc capacity \( u_{nkt}^k = \text{arc}(s, 1) \)
- Assign arc cost \( c_{nkt}^k = 0.0 \) to the arc \((t + (k - 1)T, t + (k - 1)T + 1)\)
- Create a flow variable \( h_{nkt}^k \) for the arc \((t + (k - 1)T, t + (k - 1)T + 1)\)

End For

End For

**Step 4:** For \( k \in \{1, 2, \ldots, K_n\} \)

For \( t \in \{1, 2, \ldots, T - 1\} \)
- Add an arc \((t + (k - 1)T, t + kT + \mu_n^k(t_{ns}^k) + \mu_n^k(t_{nc}^k))\)

Let denote resource agent to perform the \( k \) activity

Assign arc capacity \( u_{nkt} = \mu_n^k(r_{nak}) \) to arc \((t + (k - 1)T, t + kT + \mu_n^k(t_{ns}^k) + \mu_n^k(t_{nc}^k))\)

Assign arc cost \( c_{nkt}^k = 0.0 \) to the arc \((t + (k - 1)T, t + kT + \mu_n^k(t_{ns}^k) + \mu_n^k(t_{nc}^k))\)

Create a flow variable \( w_{ntk} \) for the arc \((t + (k - 1)T, t + (k - 1)T + 1)\)

End For

End For

**Step 5:** Add an arc from node \( S \) to node 1
- Assign arc capacity \( u_{n00}^k = m_{n0}^k(p_{n0}^k) \) to arc \((s, 1)\)
- Assign arc cost \( c_{n00}^k = 0.0 \) to the arc \((s, 1)\)
- Create a flow variable \( f_{n0} \)
- Assign flow to \( f_{n0} = m_{n0}^k(p_{n0}^k) \)

**Step 6:** For \( t \in \{1, 2, \ldots, T\} \)
- Add an arc \((K_nT + t, U)\) connecting node \( K_nT + t \) to node \( U \)
- Assign arc capacity \( u_{nkt} = m_{n0}^k(p_{n0}^k) \) to arc \((K_nT + t, U)\)
- Assign arc cost \( c_{nkt} = \Phi \) to the arc \((K_nT + t, U)\), where \( \Phi \) denotes the start time for the arc \((K_nT + t, U)\)
- Create a flow variable \( l_{nt}^k \) for the arc \((K_nT + t, U)\)

End For

As the tokens must conserve, the following capacity constraints (defined in (1)) and flow balance constraints (defined in (2)–(9)) must be satisfied:

\[
\sum_{\tau = t}^{t+\gamma_{nt}^{k-1}} w_{ntk} \leq m_{nt0}(r_{nak}) \forall k \in \{1, 2, \ldots, K_n\} \forall t \in \{1, 2, \ldots, T - \gamma_{nt}^{k-1}\}
\]  \hspace{1cm} (1)

\[
h_{nt} = h_{nt} + w_{nt} \forall k \in \{1\} \forall t \in \{2, \ldots, T\}
\]  \hspace{1cm} (2)

\[
h_{nt1} + w_{nt1} = f_{nt0}
\]  \hspace{1cm} (3)
\[ h_{n(K_n+1)t} + w_{nK_n(t-\gamma_{na}^{-1}+1)} = h_{n(K_n+1)(t+1)} + l_{n(t+1)} \forall t \in \{ \gamma_{na} - 1, \ldots, T - 1 \} \] (4)

\[ h_{nklt} + w_{n(k-1)(t-\gamma_{na}^{-1}+1)} = h_{nk(t+1)} + w_{nk(t+1)} \forall k \in \{ 2, \ldots, K_n \} \forall t \in \{ \gamma_{na}^{-1}, \ldots, T - \gamma_{na} \} \] (5)

\[ h_{nklt} = 0 \ \forall k \in \{ 2, \ldots, K_n \} \forall t \in \left\{ 1, \ldots, \sum_{k'}^{k-1} \gamma_{na}^{k'} \right\} \] (6)

\[ w_{nklt} = 0 \ \forall k \in \{ 2, \ldots, K_n \} \forall t \in \left\{ 1, \ldots, \sum_{k'}^{k-1} \gamma_{na}^{k'} \right\} \] (7)

\[ h_{nklt} = 0 \ \forall k \in \{ K_n + 1 \} \forall t \in \left\{ 1, \ldots, \sum_{k'}^{K_n} \gamma_{na}^{k'} \right\} \] (8)

\[ l_{nt} = 0 \ \forall t \in \left\{ 1, \ldots, \sum_{k'}^{K_n} \gamma_{na}^{k'} \right\} \] (9)

\[ h_{nklt}, w_{nklt} \in Z, \text{ where } Z \text{ is the set of nonnegative integers and } \gamma_{na}^{k} = \mu_{na}^{k} (t_{ns}^{k}) + \mu_{na}^{k} (t_{ns}^{k}). \]

**Definition 6.** \( DG_n(V_n, E_n) \) denotes a graph model constructed by applying the D2GT algorithm to \( \Psi_n \) and satisfies constraints (1) through (9).

Figure 11a–c show the graph models corresponding to the DTPN models in Figure 9a–c, respectively.
Note that $\Psi_n = (P_n, T_n, F_n, m_{n0}, \mu_n) = \|p_k\in\{1,2,\ldots,N\}\{\Omega_n^k\|_{\alpha\in R_n^k}A_n^k\}$, where $\Omega_n^k = (p_n^k, T_n^k, F_n^k, m_{n0}^k, \mu_n^k)$ and $p_n^k=P_n^k \cup \{p_n^k\} \cup \{p_0^k\}$. Therefore, the number of tokens, $m_{nt}(p)$, in place $p \in P_n^k$ at period $t$ can be represented by $h_{ntk}$ for $t \in \{1,2,3,\ldots,T\}$. The number of tokens, $m_{nt}(p)$, in place $p \in P_n^k$ at period $t$ can be represented by $w_{ntk}$ for $t \in \{1,2,3,\ldots,T\}$. The number of tokens $m_{nt}(r_a)$ in the resource idle place $r_a$ can be calculated by $m_{nt}(r_a) = \sum_{t=1}^{T} \{p_{n}(t)+p_{n}(t)\} + 1 w_{ntk}$. Based on the above reasoning, any reachable marking of $\Psi_n = (P_n, T_n, F_n, m_{n0}, \mu_n)$ can be represented in $DG_n(V_n, E_n)$. The following properties hold for $DG_n(V_n, E_n)$ and DTPN $\Psi_n=(P_n, T_n, F_n, m_{n0}, \mu_n)$.

**Property 1.** Each marking reachable from DTPN $\Psi_n = (P_n, T_n, F_n, m_{n0}, \mu_n)$ can be represented by $DG_n(V_n, E_n)$.

**Property 2.** Suppose $DG_n(V_n, E_n)$ satisfies constraints (1)–(9). Then there is a marking of $\Psi_n = (P_n, T_n, F_n, m_{n0}, \mu_n)$ corresponding to $DG_n(V_n, E_n)$.
Property 1 and Property 2 indicate that the dynamics of DTPN \( \Psi_n = (P_n, T_n, F_n, m_{0n}, \mu_n) \) can be analyzed based on \( DG_n(V_n, E_n) \). Conversely, a \( DG_n(V_n, E_n) \) represents an evolution trajectory of \( \Psi_n \). Based on this observation, Problem 1 defined previously to determine whether there exists a firing sequence \( S_n \) for \( \Psi_n \) such that the final transition \( t_{nf} \) of \( \Psi_n \) is the last firing sequence in \( S_n \) and is fired \( Q \) times by the deadline \( \theta \) can be stated as the following problem:

**Problem 2.** Find \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) in \( DG_n(V_n, E_n) \) subject to constraints (1)–(9) such that
\[
\sum_{t \in \{1,2,...,\theta\}} l_{nt} = Q.
\]

Instead of solving Problem 2 directly, we formulate Problem 3 below and show that the solution to Problem 3 can be used to check whether there exists a solution to Problem 2. Let \( DG_n(V_n, E_n) \) be the graph model constructed by applying the DTPN2GT algorithm to \( \Psi_n \). Problem 3 aims to find \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) in \( DG_n(V_n, E_n) \) by minimizing the objective function
\[
\sum_{t=1}^{T} \sum_{k=1}^{K_n+1} \left( c_{nkt}^h h_{nkt} \right) + \sum_{t=1}^{T} \sum_{k=1}^{K_n+1} \left( c_{nkt}^w w_{nkt} \right) + \sum_{t=1}^{T} \left( c_{nkt}^l l_{nt} \right).
\]

**Problem 3.**\( \min_{h_{nkt}, w_{nkt}, l_{nt}} \sum_{t=1}^{T} \sum_{k=1}^{K_n+1} \left( c_{nkt}^h h_{nkt} \right) + \sum_{t=1}^{T} \sum_{k=1}^{K_n+1} \left( c_{nkt}^w w_{nkt} \right) + \sum_{t=1}^{T} \left( c_{nkt}^l l_{nt} \right) \) subject to constraints (1) through (9).

The following property holds for Problem 2 and Problem 3 stated above.

**Property 3.** There exists a solution to Problem 2 if and only if there exists a solution to Problem 3 with \( l_{nt} = 0 \) for all \( t > \theta \).

**Proof.** Please refer to Appendix A for the proof of Property 4. \( \square \)

Property 3 states the feasibility condition for a product agent. To establish the feasibility condition for a configuration, we need more definitions as follows:

**Definition 7.** A process agent \( w_n \) without an upstream process agents is called an input process agent.

**Definition 8.** An input process agent \( w_n \) and a set of resource agents involved the in the multi-level contract net protocol form a feasible configuration if there exists a solution to Problem 3 with \( l_{nt} = 0 \) for all \( t > \theta \). Such an input process agent \( w_n \) is called feasible.

**Definition 9.** A process agent \( w_n \) with at least one upstream process agent is called a non-input process agent.

**Definition 10.** A non-input process agent \( w_n \) and a set of resource agents involved the in the multi-level contract net protocol form a feasible configuration if there exists a solution to Problem 3 with \( l_{nt} = 0 \) for all \( t > \theta \) and there exists at least one upstream feasible process agent with for each input place in \( P^{ki}_n \). Such a non-input process agent \( w_n \) is called feasible.

**Definition 11.** A process agent \( w_n \) that provides end products to a task agent is called an output process agent.

Based on the above definitions, it follows that the following property hold.

**Property 5.** If an output process agent \( w_n \) is feasible, there exist a set of process agents and resource agents involved the in the multi-level contract net protocol that form a feasible configuration.
7. Context Generation Algorithm

Generation of contextual information takes place after the negotiation process of multi-level contract net protocol is finished and the proposals submitted by relevant process agents have been accepted. Note that only a process agent that has found a feasible solution for Problem 3 will create a PRO message and submit it to the downstream process agent. If the PRO is accepted by the downstream process agent, the context generation algorithm will be invoked to generate the contextual information based on the solution found by the process agent. That is, only the set of process agents with their PRO message being accepted will enter the step of context generation.

Based on the models and theory developed in the previous section, the context generation algorithm will be proposed in this section.

For a process agent that has found a feasible solution for Problem 3 and submitted a proposal that has been accepted by the downstream process agent, the solution found in the course of solving Problem 3 will be used as the input to the context generation algorithm. The context generation algorithm extracts the information to generate the calendars for relevant resource agents from the solution, including start transition/end transition and timing information, etc. The information generated by the context generation algorithm is formatted in XML. Algorithms 2 shows the context generation algorithm.

Algorithms 2 Context Generation Algorithm

| Input : \( T, K_n, A_{kn}, h_{nk} \) the solution \( w_{nk} \) and \( l_{nt} \) found based on \( DG_n(V_n, E_n) \) |
| Output : Contextual Information, XML_Calendar, for each resource |

Step 1 : For each \( t \in \{1, 2, \ldots, T\} \)
   - For each \( k \in \{1, 2, \ldots, K_n\} \)
     - If \( w_{nk} > 0 \)
       - \( t_{ns} \leftarrow \) Start transition of \( A_{n}\) \( A_{n}\)
       - \( t_{ne} \leftarrow \) End transition of \( A_{n}\) \( A_{n}\)
       - Assign time slot \( [t + p_s(t_{sk}) + p_e(t_{ek})]\) to calendar of agent \( a \)
       - \( Activity.WorkflowID \leftarrow n \)
       - \( Activity.StartTransition \leftarrow t_{ns} \)
       - \( Activity.EndTransition \leftarrow t_{ne} \)
       - \( Activity.StartTime \leftarrow t \)
       - \( Calendar_a \leftarrow Calendar_a \cup Activity \)
     - End If
   - End For
- End For

Step 2 : For each \( t \in \{1, 2, \ldots, T\} \)
   - For each resource agent \( a \)
     - If \( Calendar_a \neq \Phi \)
       - Create XML file XML_Calendar_a
     - End If
   - End For
- End For

Although the XML includes the contextual information for guiding resource agents, it is not suitable for providing information to users. These xml files must be transformed into other formats to provide proper GUI for users. Therefore, a GUI transformation procedure is developed to transform the contextual information in XML format to present the information to users. Table 2 shows the two steps to transform the contextual information into GUI. In this paper, we apply eXtensible Stylesheet Language (XSL) Transformations (XSLT) to transform XML files into html files to present the contextual information to users. XSL is a styling language for XML which can be applied to design GUI [44]. The GUI transformation procedure shown in the Table 2 consists of two steps. The first step is done off-line whereas the second step is done on-line. For the first step, the XSLT code must
be designed. For the second step, we will illustrate the XSLT code by an example in the next section.

Table 2. GUI Transformation Procedure for Resource Agent.

| Input: XML_Calendar_a |
|------------------------|
| Output: Standard file for GUI |
| Step 1: (Off-Line) Design XSLT code for XML_Calendar_a |
| Step 2: (On-Line) Transform XML_Calendar_a to GUI for agent a by applying the XSLT code |

8. Results

We have implemented the solution algorithm by constructing the CPS models using WoPeD [45], invoking the library provided by CPLEX solver [46] to solve Problem 3 and develop the application in JADE [47], a FIPA compliant agent platform. In this section, we first illustrate the proposed method by applying it to two examples. There are three process agents in Example 1 whereas Example 2 is an example with six process agents. Following these two illustrative examples, we then present the results obtained by conducting several experiments to study the influence of time horizon, number of operations in a process and demand on computation time and response time to assess practicality of the proposed method.

Example 1: Consider a CPS with three process agents and five resource agents. A task with three units of products to be processes is released to the CPS. The due date of this task is 2021.1.16 AM 1:00. The CPS models of three process agents and five resource agents are described by PNML files. The PNML files (W1.pnml, W2.pnml, W3.pnml, R1.pnml, R2.pnml, R3.pnml, R4.pnml, R5.pnml) of the Cyber Physical Models are available for download at the following link:https://drive.google.com/drive/folders/1ZVL3Drvd5V_L0OeF7DalwMcAnNJGj5yv?usp=sharing (accessed on 25 January 2021).

The transitions in each process and the processing time of each operation are shown in Tables 3 and 4, respectively.

Table 3. Start Transition and End Transition in each process of Example 1.

| Process W1 | Start Transition | End Transition |
|------------|------------------|----------------|
| Ω1^1       | t1               | t2             |
| Ω1^2       | t3               | t4             |
| Ω1^3       | t5               | t6             |
| Ω2^2       | t7               | t8             |
| Ω3^1       | t9               | t10            |

Table 4. Processing Time of Each Operation of Example 1.

| Operation | Processing Time   |
|-----------|-------------------|
| A1^1      | µ_1^1(t1) + µ_1^1(t2) = 1 |
| A1^2      | µ_1^2(t3) + µ_1^2(t4) = 2 |
| A1^3      | µ_1^3(t5) + µ_1^3(t6) = 2 |
| A1^4      | µ_1^4(t7) + µ_1^4(t8) = 1 |
| A3^3      | µ_3^3(t9) + µ_3^3(t10) = 1 |

For this example, T is set according to the order due date and each time period is one minute. The proposed algorithm generates the contextual information in Extensible Markup Language (XML) format for each resource. The five XML files generated for R1 through R5 are R1_Calendar.xml, R2_Calendar.xml, R3_Calendar.xml, R4_Calendar.xml and R5_Calendar.xml, respectively. The contextual information of resources R1 through R5
are available for download at the following link: https://drive.google.com/drive/folders/1ZVL3Drvd5V_L00eF7DalwMcAnNJGj5yv?usp=sharing (accessed on 25 January 2021).

Although the contextual information has been generated in R1_Calendar.xml, R2_Calendar.xml, R3_Calendar.xml, R4_Calendar.xml and R5_Calendar.xml, these XML files are not suitable for providing information to users. These xml files must be transformed into other formats to provide proper GUI for users. In this paper, we apply eXtensible Stylesheet Language (XSL) Transformations (XSLT) to transform XML files into html files to present the contextual information to users. eXtensible Stylesheet Language is a styling language for XML. To apply XSLT to transform R1_Calendar.xml, R2_Calendar.xml, R3_Calendar.xml, R4_Calendar.xml and R5_Calendar.xml into html format, we design the XSLT code. The XSLT code we design is the file R_XSL1.txt shown in Table 5 and available for download at the following link: https://drive.google.com/drive/folders/1ZVL3Drvd5V_L00eF7DalwMcAnNJGj5yv?usp=sharing (accessed on 25 January 2021).

Table 5. The XSLT code of Example 1.

| R_XSL1.txt |
|------------|
| ```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/"
<html>
<body>
<h2>Contract of <xsl:value-of select="Calendar/ResourceID" /></h2>
<table border="1">
<tr bgcolor="#9acd32">
<th>Order_ID</th>
<th>Workflow_ID</th>
<th>ActivityNumber</th>
<th>StartTransition</th>
<th>EndTransition</th>
<th>Start_Year</th>
<th>Start_Month</th>
<th>Start_Day</th>
<th>Start_Hr</th>
<th>Start_Min</th>
<th>Quantity</th>
</tr>
<xsl:for-each select="Calendar/EstablishedContract/Contract">
<tr>
<td><xsl:value-of select="Order_ID" /></td>
<td><xsl:value-of select="Workflow_ID" /></td>
<td><xsl:value-of select="ActivityNumber" /></td>
<td><xsl:value-of select="StartTransition" /></td>
<td><xsl:value-of select="EndTransition" /></td>
<xsl:for-each select="WorkingTime/StartTime">
<td><xsl:value-of select="Year" /></td>
<td><xsl:value-of select="Month" /></td>
<td><xsl:value-of select="Day" /></td>
<td><xsl:value-of select="Hr" /></td>
<td><xsl:value-of select="Min" /></td>
</xsl:for-each>
<td><xsl:value-of select="Quantity" /></td>
</tr>
</xsl:for-each>
</table>
</body>
<html>
</xsl:template>
</xsl:stylesheet>``` |
By applying R_XSL1.txt, the R1_Calendar.xml, R2_Calendar.xml, R3_Calendar.xml, R4_Calendar.xml and R5_Calendar.xml files are transformed into html file format and can be displayed in browsers. Figure 12a–f show the calendars for all agents.

Figure 12. (a) Contract of an Order; (b) Contract of R1; (c) Contract of R2; (d) Contract of R3; (e) Contract of R4; (f) Contract of R5.
Example 2: The PNML files (W1.pnml, W2.pnml, W3.pnml, W4.pnml, W5.pnml, W6.pnml, R1.pnml, R2.pnml, R3.pnml, R4.pnml, R5.pnml, R6.pnml, R7.pnml) of the Cyber Physical Models in this example are available for download at the following link: https://drive.google.com/drive/folders/1FRUMtDsdl3rUQ4_8IvCfIS15tTJoJUHX?usp=sharing (accessed on 25 January 2021).

The proposed algorithm generates the contextual information in XML format for each resource. The XML files generated for R1 through R7 are R1_Calendar.xml, R2_Calendar.xml, R3_Calendar.xml, R4_Calendar.xml, R5_Calendar.xml, R6_Calendar.xml and R7_Calendar.xml. The contextual information of resources R1 through R7 are available for download at the following link: https://drive.google.com/drive/folders/1FRUMtDsdl3rUQ4_8IvCfIS15tTJoJUHX?usp=sharing (accessed on 25 January 2021).

Although the contextual information has been generated in R1_Calendar.xml, R2_Calendar.xml, R3_Calendar.xml, R4_Calendar.xml, R5_Calendar.xml, R6_Calendar.xml and R7_Calendar.xml, these XML files are not suitable for providing information to users. These xml files must be transformed into other formats to provide proper GUI for users. We apply XSL Transformations (XSLT) to transform XML files into html files to present the contextual information to users. eXtensible Stylesheet Language is a styling language for XML. To apply XSLT to transform R1_Calendar.xml, R2_Calendar.xml, R3_Calendar.xml, R4_Calendar.xml, R5_Calendar.xml, R6_Calendar.xml and R7_Calendar.xml into html format, we design the XSLT code. The XSLT code we design is the R_XSL2.txt file available for download at the following link: https://drive.google.com/drive/folders/1FRUMtDsdl3rUQ4_8IvCfIS15tTJoJUHX?usp=sharing (accessed on 25 January 2021).

By applying R_XSL2.txt, the R1_Calendar.xml, R2_Calendar.xml, R3_Calendar.xml, R4_Calendar.xml, R5_Calendar.xml, R6_Calendar.xml and R7_Calendar.xml files are transformed into html file format and can be display in browsers. Please refer to the files Order_Calendar_ScreenShot.doc and Resource_Calendar_ScreenShot.doc available for download at the following link: https://drive.google.com/drive/folders/1FRUMtDsdl3rUQ4_8IvCfIS15tTJoJUHX?usp=sharing (accessed on 25 January 2021).

An important performance index of context-aware information system is response time. Response time refers to the time for a system to react to a given input. In assessing effectiveness of a system, an important issue is to study how response time changes with respect to parameters of the problem or the solution method. There are three parameters used in the solution algorithm, including time horizon $T$, number of operations in a process and the demand. We perform a series of experiments to study the influence of these parameters on computation time of the solution method and the total response time.

### Computation Time and Response Time with respect to time horizon $T$

To study the connection between response time and time horizon $T$, we conduct several experiments by increasing time horizon $T$. Figure 13a shows the results by changing time horizon $T$ from around 302 to 1502. The results indicate that the computation time and total time are almost the same in this range. The results also indicate that the computation time due to optimization is less than 50% of the total time. Figure 13b shows the results by changing time horizon $T$ from around 1442 to 7202. The results indicate that the computation time and total time increase linearly in this range. The results also indicate that the computation time due to optimization is over 75% of the total time.

### Computation Time and Response Time with respect to number of operations

To study the connection between response time and number of operations, we conduct several experiments by increasing the number of operations. Figure 13c shows the results by changing the number of operations from 2 to 10. The results indicate that the computation time and total time grow with respect to the number of operations in this range. The results also indicate that the growth of computation time due to optimization is less significant than that of the total time. This implies that more time is spent on communication as the number of operations grows. Figure 13d shows the results by changing the number of operations from 12 to 20. The results indicate that the computation time and total time grow with respect to the number of operations in this range. The results also indicate
that the growth of computation time due to optimization is less significant than that of the total time. This implies that more time is spent on communication as the number of operations grows.

To study the connection between response time and time horizon $T$, we conduct several experiments by increasing time horizon $T$. Figure 13a shows the results by changing time horizon $T$ from around 302 to 1502. The results indicate that the computation time and total time are almost the same in this range. The results also indicate that the computation time due to optimization is less than 50% of the total time. Figure 13b shows the results by changing time horizon $T$ from around 1442 to 7202. The results indicate that the computation time and total time increase linearly in this range. The results also indicate that the computation time due to optimization is over 75% of the total time.

Figure 13. (a) Influence of time horizon $T$ on computation time and the total response time for medium time horizon; (b) influence of time horizon $T$ on computation time and the total response time for large time horizon; (c) influence of the number of operations on computation time and the total response time by changing the number of operations from 2 to 10; (d) influence of the number of operations on computation time and the total response time by changing the number of operations from 12 to 20; (e) Influence of demand on computation time and the total response time by changing the demand from 10 to 50.

In addition to the above experiments, we also perform experiments to study how computation time and total time is influenced by demand. We increase the demand from 10 to 50 and obtain the results in Figure 13e. Note that the time spent on optimization does not increase significantly but the total time increases significantly. This implies that solution algorithm is not sensitive to the demand.
9. Discussion

From the perspectives of modeling tools, solution methods and application domains, this study are different from previous studies. In terms of development of context-aware systems and applications, although there are a lot of studies relevant to context-aware systems and applications on engineering context-aware systems [16], design of implementation method in context-aware ecosystems [17], and context-aware computing for the IoT [15], there are only a few works on the development of context-aware information system for CPS, e.g., [18–22]. Moreover, these works focus on security issue [19], programming [20], process guidance [21] and achieving self-optimization by using context awareness [22]. The problem to generate temporal workflow contextual information for complex workflows of CPS is not addressed in these studies.

In terms of modeling tools, the DTPN models proposed in this paper extends the one defined in [41] by allowing multiple input places in the workflows of CPS. The timed model makes it possible to capture the characteristics of assembly operations in manufacturing systems and enable temporal planning to generate temporal contextual information. Although the problem to generate contextual information in workflow systems has been studied in [42], the method proposed in [42] cannot generate temporal workflow contextual information due to the use of untimed models. As a time interval is used to specify the timing for firing each transition, the time factor considered in the models of [43] in conjunction with the proposed algorithms can only be used for temporal reasoning of a task. It cannot generate context-aware scheduling information for assembly workflows in CPS. Comparing with the drawbacks of the models used in [43], the DTPN Cyber World models proposed in this paper endows the CPS with the capability to achieve effective generation of temporal context-aware information system based on optimization theory and a transformation method.

Several types of production processes can be handled in this study. These include sequential processes as well as assembly processes. An assembly process can be decomposed into a number of sub-processes with dependency between upstream sub-processes and downstream sub-processes. Each process agent is responsible for only one sub-process with sequential structure. This allows the proposed approach to handle a wide variety of production processes. The capacity constraints are described in (1). Analysis and optimization are based on the capacity constraints as well as the flow balance constraints. Flexibility of the physical system can be described in terms of the capabilities and the processing time to perform operations. The capability to perform the k-th operation of process agent w to by resource agent a is described by $A_{na} = \left( p_{nak}^{k}, r_{nak}^{k}, m_{nak}^{0}, f_{nak}^{k} \right)$. The processing time is specified in the $\mu_{nak}^{k}$ function. Limitations in material supply (from external entities) have been considered in $f_{nak} = m_{nak}^{k}(p_{nak}^{k}) = Q$ in constraint (3). So, the solution found by our proposed method will satisfy the constraints due to limitations in material supply from external entities.

To assess the efficiency of the proposed approach, the computational complexity of the proposed method is analyzed. The procedures involved in the proposed method to generate contextual information include the DTPN to Graph Transformation (DTPN2GT) Algorithm, the algorithm to solve Problem 3, the Context Generation Algorithm and the negotiation protocol used. For a given negotiation protocol, the complexity of the DTPN to Graph Transformation (DTPN2GT) Algorithm and the Context Generation Algorithm are analyzed as follows.

Note that the DTPN to Graph Transformation (DTPN2GT) Algorithm constructs a $(K_n + 1) \times T$ matrix of nodes in addition to a source node and a sink node for the graph $DG_n(V_n, E_n)$. Therefore, the total number of nodes in the graph $DG_n(V_n, E_n)$ is $(K_n + 1)T + 2$. So, the complexity to construct the nodes in the graph $DG_n(V_n, E_n)$ is $O(K_nT)$. The complexity to construct the arcs in $DG_n(V_n, E_n)$ is also $O(K_nT)$.

Consider a given graph $DG_n(V_n, E_n)$ with desired flow value Q. Note that Problem 3 is actually a minimum cost flow problem with additional constraints. It is well-known that a
The minimum cost flow problem can be solved by iteratively applying a shortest path algorithm such as the Dijkstra’s shortest path algorithm [48] or directly applying the algorithm for network flow problem [49]. The minimum cost flow problem is closely related to the problem to successfully find a number of shortest paths with total flows in these paths summing up to desired flow value $Q$. As the graph $DG_n(V_n, E_n)$ is an acyclic network with positive cost, the computational complexity is $O((K_nT)^2Q)$. Therefore, the complexity to construct and solve the minimum cost flow problem for $DG_n(V_n, E_n)$ is $O((K_nT)^2Q)$. As Problem 3 is actually a minimum cost flow problem with additional constraints. Therefore, a lower bound on the complexity to solve Problem 3 is $O((K_nT)^2Q)$.

The computational complexity of the Context Generation Algorithm is analyzed as follows.

The complexity of the nested loops in Step 1 of the Context Generation Algorithm is $O(TK_n)$. The complexity of the nested loops in Step 2 of the Context Generation Algorithm is $O(TR)$. The computational complexity of the Context Generation Algorithm is $O(TK_n + TR)$.

Therefore, the overall computational complexity for the proposed method is $O((K_nT)^2Q + TR)$.

Note that the above analysis only provides an estimate of a lower bound of computational complexity. As we apply the CPLEX solver to solve the problem, we need to verify the complexity results analyzed above. CPLEX uses linear/quadratic programming relaxation to compute bounds on the value of the optimal solution. It also computes linear constraints that cut off possible solutions which do not satisfy the discreteness constraints. Our computational experiences indicate that the computational complexity by applying CPLEX increases polynomially as the parameters grow. This results is consistent with the above lower bound of polynomial complexity.

10. Conclusions

Although Cyber-Physical Systems (CPS) provides a paradigm to accommodate frequent changes in manufacturing sector based on the states available from the IoT, modeling and managing operations of CPS for manufacturing sector are challenging issues due to the complex interactions between entities in the system. Development of an effective context-aware information management system to guide the entities in the system is a critical factor to attain the potential benefits of CPS. In this paper, a general architecture for developing context-aware guidance information system for resources/workers in CPS is presented. As entities (resources/workers) in CPS are distributed, a distributed architecture based on FIPA compliant multi-agent platform is adopted to model the interactions of entities in CPS. To model a CPS, DTPN is adopted as a tool to construct the Cyber Physical models. To generate contextual information for resources/workers in CPS, two problems are involved, (1) Configuration/Scheduling Feasibility Problem, and (2) Context Generation Problem. We formulate these two problems and combine analysis/theory from information science to derive condition for the existence of a feasible solution. To generate context-aware guidance information for resources/workers in CPS effectively, the model driven architecture concept from software engineering has been adopted to develop a methodology to implement the system. The implementation methodology highly relies on the transformation from Cyber Physical models to graph models and from graph models to contextual information in standard XML format. Generation of guidance information for workers is also done by transforming contextual information from standard XML format to the target presentation format based on XSLT.

The inputs to CPS include the Cyber World models of resources and workflows whereas the outputs are presented in standard format such as HTML used to guide resources/workers to perform operations. The proposed algorithm is applied to illustrative examples to generate contextual information to guide resources in CPS. To demonstrate practicality of the proposed method, a series of experiments have been conducted to study the influence of parameters, including time horizon, number of operations in a process.
and demand, on computation time and total response time. The results indicate that 
computation time and total response time increase polynomially with respect to these 
parameters and show that the proposed method is effective in solving real problems. One 
future research direction is to extend the method to deal with other types of workflows in 
CPS. Although we limit the scope of this paper to generation of context-aware workflow 
management information to guide the operations for resources in CPS, the information 
generated by the proposed method provides insight into feasibility to meet customers’ 
demand by a deadline, which is an important quality of service issue. An interesting 
future research direction is to apply, modify or integrate the proposed method in service 
provision systems.

Funding: This research was supported in part by the Ministry of Science and Technology, Taiwan, 
under Grant MOST 109-2410-H-324-001.

Institutional Review Board Statement: “Not applicable” for studies not involving humans or animals.

Informed Consent Statement: “Not applicable” for studies not involving humans.

Data Availability Statement: Data available in a publicly accessible repository described in the article.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A

Proof of Property 4:

Proof of only if part. Let \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) for \( G_n(V_n, E_n) \) such that \( \sum_{t \in \{1,2,\ldots,\theta\}} l_{nt} = Q \) be a solution to Problem 2. We need to prove that there exists a solution to Problem 3 with \( l_{nt} = 0 \) for all \( t > \theta \). As \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) is a solution to Problem 2, it must be a feasible solution to Problem 2. \( \Box \)

Due to conservation of flows in the network of \( G_n(V_n, E_n) \), \( t \in \{1,2,\ldots,\theta\} \) that \( \sum_{t} l_{nt} = Q \). As \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) is a feasible solution to Problem 2, \( \sum_{t \in \{1,2,\ldots,\theta\}} l_{nt} = Q \). Therefore, \( l_{nt} = 0 \) for all \( t > \theta \).

As the constraints (1)–(9) in Problem 2 and Problem 3 are the same and \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) is a feasible solution to Problem 2, \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) must be a feasible solution to Problem 3.

Next, we show that \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) is an optimal solution to Problem 3. To prove this, we show that the objective function of \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) is less than or equal to an optimal solution \( h_{nkt}^*, w_{nkt}^* \) and \( l_{nt}^* \). As \( h_{nkt}^*, w_{nkt}^* \) and \( l_{nt}^* \) is an optimal solution to Problem 3, \( h_{nkt}^*, w_{nkt}^* \) and \( l_{nt}^* \) must be a feasible solution that satisfies constraints (1)–(9).

Let \( \Gamma = \{ t \mid l_{nt}^* > 0, t \in \{\theta + 1, \theta + 2, \ldots, T\} \} \). The objective function value for \( h_{nkt}^*, w_{nkt}^* \) and \( l_{nt}^* \) is

\[
\sum_{t \in \{1,2,\ldots,\theta\}} \left( \sum_{k=1}^{K_t} c_{nkt} h_{nkt}^* + \sum_{k=1}^{K_t} c_{nkt} w_{nkt}^* + \sum_{t=1}^{T} c_{nt} l_{nt}^* \right) = \sum_{t \in \{1,2,\ldots,\theta\}} \sum_{t \in \Gamma} \left( c_{nt} l_{nt}^* \right) = \sum_{t \in \Gamma} \left( c_{nt} l_{nt}^* \right) = \Phi. \]

The objective function value for \( h_{nkt}, w_{nkt} \) and \( l_{nt} \) is

\[
\sum_{t \in \{1,2,\ldots,\theta\}} \left( \sum_{k=1}^{K_t} c_{nkt} h_{nkt} + \sum_{k=1}^{K_t} c_{nkt} w_{nkt} + \sum_{t=1}^{T} c_{nt} l_{nt} \right) = \sum_{t \in \{1,2,\ldots,\theta\}} \sum_{t \in \Gamma} \left( c_{nt} l_{nt} \right) = \sum_{t \in \Gamma} \left( c_{nt} l_{nt} \right) = 0 \leq \Phi. \]
The objective function of $h_{nkt}$, $w_{nkt}$ and $l_{nt}$ is less than or equal to an optimal solution $h_{nkt}^*, w_{nkt}^*$ and $l_{nt}^*$ to Problem 3. Therefore, $h_{nkt}$, $w_{nkt}$ and $l_{nt}$ is an optimal solution to Problem 3.

**Proof of if part.** In this part, we need to prove that if there exists a solution $h_{nkt}$, $w_{nkt}$ and $l_{nt}$ to Problem 3 with $l_{nt} = 0$ for all $t > \theta$, there exists a solution to Problem 2 with $\sum_{t \in \{1,2,...,\theta\}} l_{nt} = Q$. □

Suppose $h_{nkt}$, $w_{nkt}$ and $l_{nt}$ is a solution to Problem 3 with $l_{nt} = 0$ for all $t > \theta$. The solution $h_{nkt}$, $w_{nkt}$ and $l_{nt}$ must be a feasible solution that satisfy constraints (1)–(9) in Problem 2 and Problem 3.

Due to conservation of flows in the network of $G_n(V_n, E_n)$, the equation $\sum_{t \in \{1,2,...,T\}} l_{nt} = Q$ must be satisfied. Therefore,

$$\sum_{t \in \{1,2,...,T\}} l_{nt} = \sum_{t \in \{1,2,...,\theta\}} l_{nt} + \sum_{t \in \{\theta+1,\theta+2,...,T\}} l_{nt} = Q.$$

As $l_{nt} = 0$ for all $t > \theta$,

$$\sum_{t \in \{1,2,...,T\}} l_{nt} = \sum_{t \in \{1,2,...,\theta\}} l_{nt} + \sum_{t \in \{\theta+1,\theta+2,...,T\}} l_{nt} = \sum_{t \in \{1,2,...,\theta\}} l_{nt} = Q.$$

Therefore, $h_{nkt}$, $w_{nkt}$ and $l_{nt}$ is a solution to Problem 2 with $\sum_{t \in \{1,2,...,\theta\}} l_{nt} = Q$. Q.E.D.

**References**

1. Intelligent Manufacturing Systems. Available online: http://www.ims.org/index.html (accessed on 15 January 2021).
2. Molina, A.; Rodríguez, C.A.; Ahuett, H.; Cortés, J.A.; Ramírez, M.; Jiménez, G.; Martínez, S. Next-generation manufacturing systems: Key research issues in developing and integrating reconfigurable and intelligent machines. *Int. J. Comput. Integr. Manuf.* 2005, 18, 525–536. [CrossRef]
3. Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0, Final Report of the Industrie 4.0 Working Group. Available online: https://www.din.de/blob/76902/e8cac883812f3f428536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf (accessed on 15 January 2021).
4. Bortolini, M.; Faccio, M.; Galizia, F.G.; Gamberi, M.; Pilati, F. Adaptive Automation Assembly Systems in the Industry 4.0 Era: A Reference Framework and Full–Scale Prototype. *Appl. Sci.* 2021, 11, 1256. [CrossRef]
5. Rajkumar, R.; Lee, I.; Sha, L.; Sankovic, J. Cyber-physical systems: The next computing revolution. In Proceedings of the 47th ACM/IEEE Design Automation Conference (DAC’10), Anaheim, CA, USA, 13–18 June 2010; pp. 731–736.
6. Lee, J.; Bagheri, K.; Kao, H.-A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manuf. Lett.* 2015, 3, 18–23. [CrossRef]
7. Gershenfeld, N.; Krikorian, R.; Cohen, D. The Internet of Things. *Sci. Am.* 2004, 291, 76–81. [CrossRef]
8. Kolberg, D.; Zühlke, D. Lean Automation enabled by Industry 4.0 Technologies. *IFAC-PapersOnLine* 2015, 48, 1870–1875. [CrossRef]
9. National Institute of Standards and Technology. *Workshop Report on Foundations for Innovation in Cyber-Physical Systems*, January 2013; National Institute of Standards and Technology: Gaithersburg, MA, USA, 2013. Available online: https://www.nist.gov/system/files/documents/el/CPS-WorkshopReport-1-30-13-Final.pdf (accessed on 15 January 2021).
10. Evans, D. The Internet of Everything: How More Relevant and valuable Connections Will Change the World? Available online: https://newsroom.cisco.com/video-content?type=webcontent&articleId=1112141 (accessed on 15 January 2021).
11. Abdelwahab, S.; Hamdaoui, B.; Guizani, M.; Rayes, A. Enabling Smart Cloud Services through Remote Sensing: An Internet of Everything Enabler. *IEEE Internet Things J.* 2014, 1, 276–288. [CrossRef]
12. Kortuem, G.; Kawar, F.; Sundramoorthy, V.; Fitton, D. Smart objects as building blocks for the Internet of things. *IEEE Internet Comput.* 2010, 14, 44–51. [CrossRef]
13. Dey, A.K. Understanding and Using Context. *Pers. Ubiquitous Comput.* 2001, 5, 4–7. [CrossRef]
14. Schilit, B.N.; Theimer, M.M. Disseminating active map information to mobile hosts. *IEEE Netw.* 1994, 8, 22–32. [CrossRef]
15. Perera, C.; Zaslavsky, A.B.; Christen, P.; Georgakopoulos, D. Context Aware Computing for The Internet of Things: A Survey. *IEEE Commun. Surv. Tutor.* 2014, 16, 414–454. [CrossRef]
16. Alegre, U.; Augusto, J.C.; Clark, T. Engineering context-aware systems and applications: A survey. *J. Syst. Softw.* 2016, 117, 55–83. [CrossRef]
17. Pradeep, P.; Krishnamoorthy, S. The MOM of context-aware systems: A survey. *Comput. Commun.* 2019, 137, 44–69. [CrossRef]
18. Ivanov, R.; Weimer, J.; Lee, I. Towards Context-Aware Cyber-Physical Systems. In *2018 IEEE Workshop on Monitoring and Testing of Cyber-Physical Systems (MT-CPS)*; IEEE: Piscataway, NJ, USA, 2018; pp. 10–11. [CrossRef]
19. Wan, K.; Alagar, V. Context-Aware Security Solutions for Cyber Physical Systems. In Context-Aware Systems and Applications. ICCASA 2012. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 109; Vinh, P.C., Hung, N.M., Tung, N.T., Suzuki, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2013. [CrossRef]
20. Gaur, S.; Almeida, L.; Tovar, E.; Reddy, R. CAP: Context-Aware Programming for Cyber Physical Systems. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 1009–1016.
21. Kammerer, K.; Pryss, R.; Sommer, K.; Reichert, M. Towards Context-Aware Process Guidance in Cyber-Physical Systems with Augmented Reality. In Proceedings of the 2018 4th International Workshop on Requirements Engineering for Self-Adaptive, Collaborative, and Cyber Physical Systems (RESACS), Banff, AB, Canada, 20–20 August 2018; pp. 44–51. [CrossRef]
22. Scholze, S.; Barata, J. Context Awareness for Flexible Manufacturing Systems Using Cyber Physical Approaches. In Technological Innovation for Cyber-Physical Systems. DoCEIS 2016. IFIP Advances in Information and Communication Technology, Camarin-ha-Matos, L.M., Falcão, A.J., Vafei, N., Najdi, S., Eds.; Springer: Cham, Switzerland, 2016; Volume 470. [CrossRef]
23. Bortolini, M.; Faccio, M.; Gamberi, M.; Pilati, F. Motion Analysis System (MAS) for production and ergonomics assessment in the manufacturing processes. Comput. Ind. Eng. 2020, 139, 105485. [CrossRef]
24. Azzi, A.; Battini, D.; Faccio, M.; Persona, A.; Sgarbossa, F. Inventory holding costs measurement: A multi-case study. Int. J. Logist. Manag. 2014, 25, 109–132. [CrossRef]
25. Azzi, A.; Faccio, M.; Persona, A.; Sgarbossa, F. Lot splitting scheduling procedure for makespan reduction and machine capacity increase in a hybrid flow shop with batch production. Int. J. Adv. Manuf. Technol. 2012, 59, 775–786. [CrossRef]
26. Faccio, M.; Bottin, M.; Rosati, G. Collaborative and traditional robotic assembly: A comparison model. Int. J. Adv. Manuf. Technol. 2019, 102, 1355–1372. [CrossRef]
27. Azzi, A.; Battini, D.; Faccio, M.; Persona, A. Mixed model assembly system with multiple secondary feeder lines: Layout design and balancing procedure for ATO environment. Int. J. Prod. Res. 2012, 50, 5132–5151. [CrossRef]
28. Bortolini, M.; Faccio, M.; Galizia, F.G.; Gamberi, M.; Pilati, F. Design, engineering and testing of an innovative adaptive automation assembly system. Assem. Autom. 2020, 40, 531–540. [CrossRef]
29. Bortolini, M.; Galizia, F.G.; Mora, C.; Pilati, F. Reconfigurability in cellular manufacturing systems: A design model and multi-scenario analysis. Int. J. Adv. Manuf. Technol. 2019, 104, 4387–4397. [CrossRef]
30. Müller, T.; Jazdi, N.; Schmidt, J.P.; Weyrich, M. Cyber-Physical Production Systems: Enhancement with a Self-Organized Reconfiguration-Management. 2020. Available online: https://www.researchgate.net/publication/341321831_Cyber-Physical-Production_Systems_enhancement_with_a_self-organized_reconfiguration_management (accessed on 15 January 2020).
31. Yelles-Chaouche, A.R.; Gurevsky, E.; Brahim, N.; Dolgui, A. Reconfigurable manufacturing systems from an optimisation perspective: A focused review of literature. Int. J. Prod. Res. 2020, 1–19. [CrossRef]
32. Brandimarte, P.; Villa, A. (Eds.) Optimization Models and Concepts in Production Management; Gordon & Breach: Basel, Switzerland, 1995.
33. Law, A.M.; Kelton, W.D. Simulation Modeling and Analysis, 2nd ed.; McGraw-Hill: New York, NY, USA, 1991.
34. Peck, S.; Dicesare, F.; Harhalakis, G.; Proth, J.M.; Silva, M.; Vernadat, F.B. Practice of Petri Nets in Manufacturing. J. Oper. Res. Soc. 1994, 45, 1094. [CrossRef]
35. Monostori, L.; Váncza, J.; Kumara, S. Agent-Based Systems for Manufacturing. CIRP Ann. 2006, 55, 697–720. [CrossRef]
36. Smith, R.G. The Contract net protocol: High-level communication and control in a distributed problem solver. IEEE Trans. Comput. 1980, 29, 1104–1113. [CrossRef]
37. Murata, T. Petri nets: Properties, analysis and applications. Proc. IEEE 1989, 77, 541–580. [CrossRef]
38. Billington, J.; Christensen, S.; Van Hee, K.; Kindler, E.; Kummer, O.; Petrucci, L.; Post, R.; Stehno, C.; Weber, M. The Petri Net Markup Language: Concepts, Technology, and Tools. Constr. Side Channel Anal. Secur. Des. 2003, 2679, 483–505. [CrossRef]
39. Hsieh, F.-S.; Lin, J.-B. A dynamic scheme for scheduling complex tasks in manufacturing systems based on collaboration of agents. Appl. Intell. 2014, 41, 366–382. [CrossRef]
40. Hsieh, F.-S. A hybrid and scalable multi-agent approach for patient scheduling based on Petri net models. Appl. Intel. 2017, 47, 1068–1086. [CrossRef]
41. Hsieh, F.-S. Design of scalable agent-based reconfigurable manufacturing systems with Petri nets. Int. J. Comput. Integr. Manuf. 2017, 31, 748–759. [CrossRef]
42. Hsieh, F.-S.; Lin, J.-B. Development of context-aware workflow systems based on Petri Net Markup Language. Comput. Stand. Interfaces 2014, 36, 672–685. [CrossRef]
43. Hsieh, F.-S.; Lin, J.-B. Context-aware workflow management for virtual enterprises based on coordination of agents. Int. J. Manuf. 2012, 25, 393–412. [CrossRef]
44. Extensible Stylesheet Language. Available online: https://www.w3schools.com/xml/xsl_intro.asp (accessed on 15 January 2021).
45. Workflow Petri Net Designer (WoPeD). Available online: https://woped.dbw-karlsruhe.de/ (accessed on 15 January 2021).
46. IBM CPLEX Optimizer. Available online: https://www.ibm.com/analytics/cplex-optimizer (accessed on 15 January 2021).
47. Java Agent DEvelopment Framework (JADE). Available online: https://jade.tilab.com/ (accessed on 15 January 2021).
48. Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1959, 1, 269–271. [CrossRef]
49. Edmonds, J.S.; Karp, R.M. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. J. ACM 1972, 19, 248–264. [CrossRef]