COMPUTER SCIENCE | RESEARCH ARTICLE

Review on Petri Net Modeling and Analysis of a Smartphone Manufacturing System

Yi-Nan Lin¹, Tsang-Yen Hsieh³, Cheng-Ying Yang², Victor R.L. Shen³,⁴*, Tony Tong-Ying Juang⁴ and Ting-Jui Huang⁴

Abstract: Currently, the marketing strategy in the manufacturing industry is becoming more and more competitive. The emergence of lean production aims to reduce unnecessary wastes in the manufacturing processes. In order to achieve this goal, it is necessary to optimize manufacturing processes so as to reduce their costs. Therefore, reducing manufacturing costs has become the main developing direction people are now pursuing. The iPhone XS developed by Apple is now a popular product. To this end, this paper aims to describe its manufacturing system by using Petri nets, and to integrate the Prim algorithm with a minimum spanning tree (MST) to find the minimal cost path, achieving the goal of reducing the manufacturing costs. Two simulation software tools, WoPeD and PIPE, were applied to conduct the experiments. The experimental results have significantly indicated that the proposed approach is more feasible and acceptable than others.

Subjects: Automation Control; Petri Net Theory; Concurrent Computing; Systems Engineering

Keywords: iPhone XS; Petri net modeling; minimum spanning tree; Petri net tools; simulation

ABOUT THE AUTHOR

Yi-Nan Lin received his B.S. degree in Electrical Engineering from National Taiwan Institute of Technology, Taipei City, in 1989; M.S. degree in Computer Science & Engineering from Yuan Ze University, Tao-Yuan City, Taiwan, in 2000; and a Ph.D. degree in Electrical Engineering from Chang Gung University, Tao-Yuan City, Taiwan, in 2009. Since 1990, he has joined the Department of Electrical Engineering at Ming Chi University of Technology (MCUT), Taishan District, New Taipei City, Taiwan. He is now an Associate Professor at the Department of Electronic Engineering in MCUT. His current research interests include Petri net theory and applications, systems engineering, error-control coding, digital transmission systems, and embedded system design.

PUBLIC INTEREST STATEMENT

Petri nets are a graphical and mathematical model widely applied in the field of systems engineering. It is a promising tool for describing information systems that are characteristic of concurrency, parallelism, distribution, nondeterminism, probability, time, fuzziness, and so on. As a graphical model, Petri nets can be used to serve as a visualization aid similar to flowcharts. However, the flowcharts have no all the above-mentioned characteristics. In this article, the manufacturing processes are modeled by Petri nets for the purpose of optimizing and reducing manufacturing costs. Meanwhile, two Petri net simulation software tools, WoPeD and PIPE, are open sources, which are free of charge and easy to be used. Thus it is the main reason for modeling and analyzing the smart manufacturing systems by using Petri nets from the view-points of practical applications.
1. Introduction
The marketing strategy in the manufacturing industry is truly becoming more and more competitive. In the traditional production, single product and mass production are employed to reduce cost, but it actually leads to a large amount of inventory cost. Today’s consumers are eager for the diversified demands, and the single product is no longer satisfied (Karimi & Nickpayam, 2017). Moreover, the traditional mass production no longer satisfies consumers’ requirements because the demands of consumers are versatile. Hence, the traditional mass production mode is not suitable for the current market competition (Shahbaz et al., 2019). However, the lean production is to seek just-in-time (JIT) manufacturing and to reduce unnecessary wastes in the manufacturing processes. Therefore, the lean production has become an important issue for today’s enterprises (Shahbaz et al., 2019).

The emergence of lean production intends to reduce unnecessary wastes in the manufacturing processes. In order to achieve this goal, it is necessary to optimize manufacturing processes so as to reduce costs. The advantages of lean production are to introduce automatic production technology and to improve production efficiency (F. Yang et al., 2014). Petri nets were used to optimize the manufacturing processes and to effectively achieve the goal of lean production. For the cost reduction, the lean production introduces smart manufacturing systems, which could effectively increase production efficiency. Petri nets were used to characterize the manufacturing processes by using the Prim algorithm with a minimum spanning tree (MST) to seek a minimum cost path (Prim Algorithm, 2019). For the purpose of clearly explaining the feasibility of the proposed approach, Apple iPhone XS is chosen as the target manufacturing product.

Hereby, the concept of Petri nets is applied to study the manufacturing processes of the iPhone XS. Thus, this paper aims to analyze the manufacturing processes using Petri net tools, WoPeD and PIPE, so as to overcome the disadvantages of the traditional mass production and to enhance the competitiveness of the enterprises (Wang et al., 2019).

In Section 2, an amount of literature that is about to analyze the workflow based on Petri nets is discussed. In Section 3, the method regarding how to minimize the production cost based on Petri nets is detailed. In Section 4, the target iPhone XS’s components and manufacturing processes are converted to Petri nets so as to reach the goal of cost optimization. In Section 5, the conclusion and the future prospects are remarked.

2. Literature review
This Section presents the basic characteristics and applications of Petri nets, and how to optimize the production cost.

2.1. Petri nets
Petri net (PN) is a graphical and mathematical model. It is a discrete event dynamic system that can be applied in many systems. PN can be used to describe a series of systems, such as parallel, asynchronous, concurrent, nondeterministic, stochastic, and distributed systems. Its graphical representation is suitable for the workflow field modeling, making the model look intuitive and easier to understand. It can also test, analyze, and solve some problems (Chen et al., 2014).

Petri net is a simple model, which consists of many elements including place, transition, arc, and token. A set of places is denoted by $P = \{p_1, p_2, \ldots, p_n\}$, and a set of transitions is denoted by $T = \{t_1, t_2, \ldots, t_m\}$. If represented in a graphical model, a circle $\bigcirc$ denotes a place and a rectangle $\blacksquare$ denotes a transition. An arrow is an arc, which is directional and has a weight on it. A directed arc represents the connection of a place to a transition, some arcs are from transitions to places, and other arcs are from places to transitions. A set of arcs is denoted by $A = \{(p_i, t_j), (t_s, p_j)\}$, and $w = (p_i, t_j)$ is the weight from place $i$ to transition $j$. 
An assignment of tokens to the places of a Petri net is denoted as marking $M$. $M(p_i)$ denotes the number of tokens in place $i$. Token is a dynamic object in a place that can be moved from one place to another. When the Petri net is in execution, the position and number of tokens will be changed. So, the token is used to represent the execution of a Petri net. The changes of a token can describe various activities. Thus, Petri nets can be applied in many fields. Furthermore, in order to satisfy specific requirements, various restrictions can be added. The dynamic system state change of the process is described by the token firing and moving in the PN model. When each input place has one or more tokens, the transition will be enabled. When the transition is enabled, the transition may or may not be fired. When a transition is fired, the token will be disappeared in the input place and be transferred to the output place. In the PN model, the small dots • are denoted as tokens. Figure 1 shows a simple Petri net model. In Figure 1(a), the places $P = \{p_1, p_2, p_3\}$, the transition $T = \{t_1\}$, the arcs $A = \{(p_1, t_1), (p_2, t_1), (t_1, p_3)\}$, the marking $M = \{M(p_1), M(p_2), M(p_3)\} = \{2 \ 2 \ 0\}$, and the weights $w(p_1, t_1) = 2$, $w(p_2, t_1) = 1$, $w(t_1, p_3) = 2$. After firing $t_1$, the status is shown in Figure 1(b).

2.2. Basic properties of Petri net

The Petri net model has several properties such as liveness, reachability, boundedness, reversibility, and coverability (Murata, 1989; Rezig et al., 2015; Tebani et al., 2018).

2.2.1. Liveness
If a Petri net model starts from $M_0$, regardless of reaching any marking $M$ that can make any enabled transition be fired. Discussing the liveness of a Petri net, the key idea is to avoid the problem of a deadlock. A deadlock in Petri nets means that no transition will be fired at all. A Petri net with liveness can make sure that under any marking, one can still find a transition that will be fired. In fact, the characteristics of liveness not only avoid the problem of a deadlock, but also prevent the stagnation in any part of the Petri net model.

The levels of liveness are categorized as follows:

L0: Dead, unable to fire any transition.
L1: The transition is fired once at least.

L2: The transition is fired k times at least.

L3: The transition is fired unlimited times.

L4: All firing sequences are L1.

2.2.2. Reachability
If it exists a sequence of transition firing that starts from the initial marking \( M_0 \), going through all reachable locations to reach the final marking \( M_n \), it is called reachability.

\[ M \in R(M_0). \]

where \( X \) denotes a transition vector and \( A = \text{Incidence Matrix} = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \\ t_1 & 1 & 0 & 0 & 0 \\ t_2 & -1 & -1 & 1 & 0 \\ t_3 & 0 & 0 & -1 & 1 \\ t_4 & 0 & 0 & 0 & -1 \end{bmatrix} \]

2.2.3. Boundedness
Boundedness means that the number of tokens cannot exceed a certain value. If the number of tokens does not exceed the value of \( k \), the place in a Petri net is called \( k \)-bounded. If it exceeds the value of \( k \), it is said to be overflow. Besides, if it is 1-bounded, it is said to be safe.

2.2.4. Reversibility
A Petri net is reversible if it exists any marking \( M \) that can return to the initial marking \( M_0 \).

\[ \forall M \in R(M_0) \]

2.2.5. Coverability
Any marking \( M \) in a Petri net is coverable if it exists \( M_1 \) in \( R(M_0) \) such that \( M_1(p) \geq M(p) \) for each \( p \) in the net.

\[ \forall M_1 \in R(M_0), M_1(p) \geq M(p), \forall p \in P \]

2.3. Cost optimization by minimum spanning tree
The algorithm of minimum spanning tree (MST) can be used for cost optimization. A well-known one is called the Prim algorithm, which is used to find a representative shortest path. It divides all vertices into two groups. The first group is a set of vertices \( S \) in which the shortest path has been found, and the second group is a set of vertices \( U \) which is remaining undetermined shortest paths. According to the shortest path, the second set of vertices is added to \( S \) in the order from small weight to large weight. Until all vertices join \( S \), it means that the vertices are all connected.

2.4. Related work
Many subclasses of Petri nets are described as follows: A subclass of generalized Petri nets was used to model a flexible manufacturing system as a livelock control policy (Hou et al., 2014). Deterministic timed weighted marked graphs were developed for the purpose of system
An optimization approach was presented to improve the Petri net controller design for automated manufacturing systems (Hu, Zhou, Li, Tang et al., 2013a). Modeling and optimization of hybrid systems were proposed for the tweeting factory (Lennartson et al., 2016). A logistic support architecture with Petri net was designed in a cloud environment for services and profit optimization (Jiang et al., 2016). Fuzzy Petri nets were proposed for novel shot boundary detection in news streams (S.-H. Yang et al., 2019).

3. The proposed approach
This paper uses WoPeD and PIPE as Petri net's simulation software tools to find the minimum cost. Through WoPeD, we can also check whether a deadlock in the system occurs or not and find the
location of a deadlock (Hu, Zhou, Li, Tang et al., 2013a; Shen et al., 2018; Hu et al., 2013b; Souravlas and Roumeliotis, 2015). Through PIPE, it is intended to find the minimum cost path.

3.1. System flowchart
The system flowchart of this paper is shown in Figure 2. First, transform the iPhone XS manufacturing processes into Petri net models. Second, look up each component cost in the vendors’ directory and build the related costs list. Third, assemble each component to construct a system module. Fourth, based on the Petri net models, the Prim algorithm is used to find the MST. Fifth, the MST represents the minimal cost path. Keep on going until all the system modules are finished.

3.2. WoPeD and PIPE
WoPeD is a Petri net open-source software tool, which can be used to model, simulate, and analyze the workflow processes (WoPeD, 2019).

PIPE is also a Petri net open-source software tool, which can provide a complete set of analysis tools to simulate the relationship of project activities, and to generate the performance statistics for simulation results (PIPE, 2019).

In this paper, we use WoPeD to find where the location of a deadlock is, and use PIPE to get where the minimum siphons are. We use two simulation software tools because PIPE cannot be used to clearly mark where the dead transition is and where the actual location of a deadlock is. However, WoPeD cannot be used to find the minimum siphons, and cannot be used to know where a siphon is formed.

3.3. Cost optimization by Prim algorithm
Prim algorithm (Prim Algorithm, 2019) is described as follows:

Input: graph structure.

A weighted and connected graph, where \( V \) denotes a set of vertices, and \( E \) denotes a set of edges.

Output: minimum spanning tree (MST).

Procedure:

Step 1:
Initialization: \( V_{\text{new}} = (x) \), where \( x \) is any node in the set \( V \) for a starting node. \( E_{\text{new}} = \emptyset \).

Step 2:
Repeat the following steps until \( V_{\text{new}} = V \):

a. Select the smallest weight edge \((u, v)\) in set \(E\), where \(u\) is an element in the set \(V_{\text{new}}\), and \(v\) is a vertex in \(V\) that is not added to \(V_{\text{new}}\) (if there are multiple edges satisfying the conditions above, namely, the same-weighted edge, then choose any one of them).
b. Add v to set $V_{new}$ and add $(u, v)$ to set $E_{new}$.

**Step 3:**

Use sets $V_{new}$ and $E_{new}$ to describe the minimum spanning tree (MST).

### 3.4. Illustrative example

*Figure 3* is a weighted and connected graph. The numbers on the edge represent weights.

In *Figure 4*, pick vertex A as a start point. Vertices B, C, and D are connected to vertex A, respectively. Among them, vertex D is the one closest to vertex A. Therefore, vertex D and its corresponding edge AD are represented by a bold line.

In *Figure 5*, the next vertex is the one closest to A or D. The distance between B and A is 7, the distance between C and A is 5, and the distance between C and D is 2, the distance between E and D is 5. Therefore, vertex C is the one closest to A or D; thus, vertex C and its corresponding edge DC is represented by a bold line.

In *Figure 6*, the adjacent vertices to vertices A, D, and C are vertices B, F, and E. The closest vertex to vertices A, D, and C is vertex E, i.e. the distance is 3. Vertex E and its corresponding edge CE are represented by a bold line.

In *Figure 7*, the vertex adjacent to A, D, C, and E are B and F. The distance between A and B is 7, the distance between C and B is 6, the distance between C and F is 8, and the distance between E and F is 5. Hence, the closet vertex is vertex F, i.e. the distance is 5. Thus, vertex F and its corresponding edge EF is represented by a bold line.

In *Figure 8*, B is the only vertex that has not been selected, whose distance from A is 7, from C is 6, and from F is 2. Hence, vertex B and its corresponding edge FB is represented by a bold line. Now, all
Table 1. Preliminary bill-of-materials summary for iPhone XS

| Components                          | Cost  |
|-------------------------------------|-------|
| Display/Touchscreen Module          | $110.00 |
| Major Mechanical/Electro-Mechanical Elements | $61.00 |
| Cameras                             | $35.00 |
| Apps Processor                      | $27.50 |
| RF Chipset (includes the following) | $18.00 |
| Power Management ICs                | $14.25 |
| Memory                              | $33.45 |
| RF/PA Section                       | $16.60 |
| User Interface IC Content           | $10.05 |
| Sensors                             | $2.35  |
| True Depth Sensing Suite            | $16.70 |
| WLAN/BT Module                      | $7.35  |
| Battery Pack(s)                     | $6.00  |
| Box Contents/Others                 | $12.00 |

the vertices have been selected. The bold line represents a minimum spanning tree. In this example, if the weight represents the cost, the minimum spanning tree is the minimum cost path, which is 15.

4. Experimental results
The target product of this study is iPhone XS (The 999 USD iPhone X, Oct. 2017; iPhone XS Max, 2018; Murai, 2018). We convert the actual costs of the iPhone XS into Petri nets to find the minimum cost (Dwyer et al., 2015; Hu & Zhou, 2015).

4.1. Product supply chain
This paper uses the PN model to simulate the components of the iPhone XS and its supply chain of the ordering system (Shahbaz et al., 2019; Shaikh et al., 2019). Prim algorithm was used to optimize the manufacturing processes in order to minimize the production cost.

Figure 9 shows the components of iPhone XS, including cables, PCB for the loudspeaker, microphone, ambient light sensor, flood illuminator and proximity sensor, logic board 1, logic board 2, front/face ID camera array, wireless charging coil, lighting port cable, logic board 3, rear ISMP camera, flash, taptic engine, barometric vent, loudspeaker, SIM tray, front/display, battery, rear enclosure (iPhone XS, Dec. 2018; Teardown: Apple iPhone X, Dec., 2018).

In Figure 9, the components of the iPhone XS are very diverse. In order to make the supply chain be more relevant to today’s manufacturing industry, this paper simulated the supply chain under a variety of vendors. The competition from different vendors in each component makes the supply chain more complex. Therefore, the status of different vendors is integrated into PNs. Moreover, select the right vendor and technology to determine the final minimum production cost. Table 1 is the bill-of-materials (BOM) summary for iPhone XS cited from the Apple website, which shows the cost of each component (Hao, 2008; Uzam et al., 2013; Zhang et al., 2011).

4.2. The iPhone XS-like manufacturing processes with Petri nets
Figures 10, 11, and 12 show the manufacturing processes of iPhone XS with PNs (Jiang et al., 2016). The structure of this PN is a tree structure, and each branch in the tree represents the components of the iPhone XS. Finally, all branches are combined to become a complete iPhone XS. In the PN model, tokens go through each enabled branch to the bottom destination iPhone XS to represent the full production cycle. The weight on the arc represents the number of tokens required for the
production cost. The production simulation for each module is described in Figures 13–21 including the assumed yield and various production conditions (He et al., 2017; Latorre Biel et al., 2011; Rodriguez et al., 2013).

Figure 13 shows the PN of the iPhone XS display module. The starting node represents the funds, and reaches the terminal node via three paths of three vendors including A, B, and C. The leftmost path has enough funds of 80 tokens of production cost to trigger T0, and arrives at the location of vendor A. Then, trigger T3 to complete the display module and transfer the tokens to the terminal node of the display module.

The intermediate path triggers T2 to reach the location of vendor B with 20 tokens of production cost, and then fire T4 to complete one part of the display module. However, before triggering T12 to transfer the token to the terminal node of the display module, it needs support from vendor C to complete another part of the display module. While the 1/2 and 2/2 parts on the display module have enough tokens, t12 will successfully trigger the terminal node of the display module.
The rightmost path triggers T7 to reach vendor C, in which the production cost is 70 tokens. Then, there are two branch paths. One branch path triggers T9 to complete the display module 2/2 part, and combines with the display module 1/2 part completed by vendor B to finish the entire display module. Then, trigger T12 to complete the display module. The other branch path is to trigger T10 to directly complete the display module to the terminal node, but it will encounter yield problems, assuming yield rate to be 90%.

In Figure 13, there are three conditions for triggering T11, 90% yield rate, return, and 10% additional cost. Because it only has a 90% yield rate, 10% additional cost is required to catch up with other production lines. Herein, a return mechanism is needed to prevent the already paid 10% additional cost when the 90% yield has not been triggered. If the return mechanism does not exist, and when the T13 is triggered, 10% of the additional cost has been paid. Without the existence of a return mechanism, T14 can be triggered directly, which means 10% of the additional cost, and refunded without triggering the 90% yield.
Therefore, the return mechanism is to prevent 10% additional cost from being refunded without triggering the 90% yield rate. In the part of vendor C, there is also a recycling mechanism T8, which is used to recover the remaining unused tokens. If there are 75 Tokens in the initial capital state, when T7 is triggered, the location of vendor C will have 70 tokens. After that, T9 will be triggered, and the location of vendor C will have 15 tokens remained, while the capital position is only remaining 5 tokens. Resulting in unavailable transitions in PN can trigger and then cannot carry out other production lines. However, through the T8 recycling mechanism, the remained 15 tokens of vendor C can be recovered and invested in the capital position. Therefore, there are 20 Tokens sufficient to trigger the T2 path.

The PN of the touch screen is shown in Figure 14. There are three main paths for three vendors A, B, and C. The leftmost path is vendor A, which needs 19 tokens of production cost. However, before completing the touch screen module, it needs support from vendor B with six tokens. Both vendors A and B need to provide enough number of tokens to trigger T1 and to transfer the tokens to the terminal node of the touch screen.

The intermediate path triggers T2 to reach vendor B. The production cost is 30 tokens, which have two branch paths. One branch path is to give vendor A support to trigger T1, and to finish the touch screen module production. The other branch path is to trigger T4 to directly complete the touch screen module, and reach the terminal node. There is a recycling mechanism T3, which can recover the remained unused tokens.
For example, 40 tokens in the initial capital state trigger T2, and vendor B has 30 tokens. Giving vendor A 6 tokens of support, the location of vendor B remains 24 tokens. At this time, the starting node has only 10 remaining tokens. In this state, the PN has no other transitions being triggered. Nevertheless, if it adds the recovery mechanism T3, the remained 24 tokens with vendor B are recycled to the capital position. Therefore, there are 34 tokens in the capital position. It can trigger T0 or T5 and T8.
The rightmost path triggers T5 to reach the touchscreen for vendor C, whose production cost is 20 tokens. Herein, considering the yield rate of vendor C, the yield is 80%; thus, the additional production cost is 20%, and two tokens are required. There is a return mechanism, in order to prevent a 20% additional cost from being refunded without triggering an 80% yield rate.

The PN of the audio codec module is shown in Figure 15. There are two main paths to reach the terminal node, including vendors A and B. The path to vendor A needs five tokens of production cost, and there are two paths. One path is to trigger T1, but it will encounter a yield problem with an 80% yield rate. Thus, the additional cost is 20%. There is a return mechanism, in order to prevent a 20% additional cost from being refunded without triggering the 80% yield rate. The other path is to give vendor B support to trigger T2. There is a recycling mechanism T6, which can recycle the unused tokens. The rightmost path triggers T0 to vendor B. However, it needs three tokens of support from vendor A before triggering T2 to complete the audio codec module.

The PN of the Near Field Communication (NFC) controller module is shown in Figure 16. There are two paths to finish the NFC controller module. The leftmost path is to trigger T0 to vendor A, which needs two tokens of production cost. Then, trigger T2 to directly finish the NFC controller module. The rightmost path needs one token to trigger T1 to reach vendor B. After reaching vendor B, trigger T3 to finish the module.

The PN of the touchscreen control module is shown in Figure 17. The leftmost path is to trigger T0 with 2 tokens to reach vendor A. Then, trigger T4 to complete the 1/2 part of the touchscreen
Figure 16. The supply chain of NFC controller module in Petri net.

Figure 17. The supply chain of touchscreen control module in Petri net.
control module. Herein, before triggering T7 to finish the touchscreen control, the support from vendor B is needed.

The intermediate path triggers T1 to reach vendor B, with three tokens of production cost. There are two paths for vendor B. One path is to trigger T5, finishing the 2/2 part of the touchscreen control module. The other path is to trigger T6 with three tokens. However, T6 needs support from vendor C before triggering. There is a recycling mechanism in vendor B. T2 recycles the unused tokens.

The rightmost path triggers T3 to reach vendor C, which has two branch paths. One branch is to trigger T6 with the support from vendor B, completing the touchscreen control module. The other branch is to trigger T8 with two tokens to complete the touchscreen control module directly.

The PN of the Bit Torrent (BT) module is shown in Figure 18. The leftmost path triggers T0 to reach vendor A. Vendor A can finish a part of the BT module. The rightmost path triggers T1 to reach vendor B. Vendor B can finish another part of the BT module. Transition T4 needs the support from vendors A and B to trigger. Finally, trigger T4 to complete the BT module.

The PN of the Frequency Modulation (FM) module is shown in Figure 19. The path to vendor A is to trigger T2 with three tokens. Before firing T6, the support from vendor B is needed. After firing T6, it will encounter an 80% yield problem. Twenty percent additional cost with two tokens will be added. There is a return mechanism, preventing a 20% additional cost from being returned without triggering an 80% yield rate.
Figure 19. The supply chain of FM module in Petri net.

Figure 20. The supply chain of WLAN module in Petri net.
The intermediate path is to trigger T3 to reach vendor B. The production cost is two tokens, which have two paths. One path is to provide support to vendor A, in order to trigger T6. The other path is to trigger T7, in order to finish 1/2 part of FM. Before firing T10, it needs to await vendor C by finishing the 2/2 part of FM. There is a recycling mechanism in vendor B. Transition T4 recycles the unused tokens.

The rightmost path is to trigger T5 to reach vendor C with three tokens. Trigger T8 to finish the 2/2 part of FM. After 1/2 part and 2/2 part of FM are combined, trigger T10 to complete the FM module.

The PN of the Wireless Local Area Networks (WLAN) module is shown in Figure 20. The WLAN module only has one vendor and one path. The path is to trigger T0 to reach the vendor. Then, trigger T1 to finish the WLAN module directly.

The PN of the Global Positioning System (GPS) module is shown in Figure 21. There are two paths to complete the GPS module. The leftmost path triggers T0 to reach vendor A, which has two branch paths. One branch triggers T2 to finish the GPS module directly. The other branch is to provide support to vendor B. The rightmost path triggers T1 to reach vendor B, which needs the support from vendor A to trigger T3. After firing T3, the GPS module is completed.

4.3. Experiments and discussion
Assemble all the iPhone XS components. The weight on the arc represents the number of tokens required for the production cost. Therefore, it is a connected and weighted graph (Dotoli et al., 2009; Hu, Zhou, Li, Tang et al., 2013a). As a result, we can find a minimum spanning tree, which represents the minimum cost. Thus, based on the above-mentioned method we find the...
Figure 22. The MST of the display module.
Figure 23. The MST of the user interface module.
The minimum critical path, which is indicated by bold-red lines. It represents the minimum cost path to finish manufacturing the product (Andolfatto et al., 2014; Lennartson et al., 2016). The cost paths for MST printed in red are shown in Figures 22, 23, and 24. In order to illustrate the experimental results, the selected modules include the display, user interface, and BT/WLAN.
Table 2. Summary of display, user interface, and BT/WLAN modules

| Items Modules | Total weight in MST | Components or Suppliers |
|---------------|---------------------|-------------------------|
| Display       | 299                 | Display-Vendors B & C,  |
|               |                     | Display Parts 1/2 & 2/ |
|               |                     | 2; Touchscreen-Vendor C,|
|               |                     | Yield, Return           |
| User interface| 52                  | Audio Codec, Vendors A &B, |
|               |                     | NFC Controller, Vendor B, |
|               |                     | Touchscreen Controller, |
|               |                     | Vendors C, Touchscreen Control |
| BT/WLAN       | 57                  | BT-Vendors A & B, BT-Ports 1/2 & 2/2; FM- |
|               |                     | Vendors B & C, FM-Parts 1/2 & 2/2; WLAN, |
|               |                     | Vendor, GPS, Vendors A & B |

Table 2 presents the components of each module and total weight in MST.

In summary, if we add a value on each arc, the value is called weight, and this graph is called a weighted graph. If the weight stands for cost, this graph is called a network. According to the definition of a spanning tree, a graph can have more than one spanning tree, and there exists an MST in the network which stands for the minimal cost. Also, according to the Prim algorithm, add all the required nodes (such as capital and finished goods), connect all the nodes, and then remove the redundant leaves and branches to get the MST. Thus, the cost optimization for each module is completed.

5. Conclusion

In this paper, Petri nets were used to describe the manufacturing processes of the iPhone XS supply chain. Using PNs to model the entire workflow, it can effectively improve the integration and efficiency of the manufacturing processes. The manufacturing cost is added to the weight of each arc. Besides, the Prim algorithm was used to find the minimum spanning tree, which represents the minimum critical cost path. Therefore, the minimum cost decision can be made on achieving the goal of improving manufacturing efficiency and earning cost optimization. The contributions of this paper are presented as follows: (1) to describe the manufacturing processes by using PNs, providing a more intuitive way to view the entire workflow; (2) to use Prim algorithm to find the minimum cost path so as to achieve the goal of manufacturing cost optimization; (3) to make the enterprises more competitive, helping the enterprises produce products more efficiently; (4) to promote the PN simulation tools, WoPeD and PIPE, with use of ease. Hopefully, the proposed approach will be applied in other fields in the future. It can also be combined with other research results in order to develop a better method for cost optimization. Furthermore, this study will be applied to many other enterprises that face a competition crisis to make them more robust in the global manufacturing processes.

Acknowledgments

The authors are very grateful to the anonymous reviewers for their constructive comments which have improved the quality of this paper.

Funding

This work was supported by the Ministry of Science and Technology, Taiwan, ROC, under grants MOST 107-2221-E-845-001-MY3 and MOST 107-2221-E-845-002-MY3.

Competing interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Author details

Yi-Nan Lin1
E-mail: jinlin@mail.mcut.edu.tw
Tsang-Yen Hsieh1
E-mail: tyhsieh001234@gmail.com
Cheng-Ying Yang2
E-mail: cyang@utaipei.edu.tw

Victor R.L. Shen3,4
E-mail: rlshen@mail.ntpu.edu.tw
E-mail: victor.rshen@mnsa.hinet.net
Tony Tong-Ying Juang4
E-mail: juang@mail.ntpu.edu.tw
Ting-Jui Huang4
E-mail: sj11060701@gmail.com

1 Department of Electronic Engineering, Ming Chi University of Technology, New Taipei City, Taiwan, ROC.
2 Department of Computer Science, University of Taipei, Taipei City, Taiwan, ROC.
3 Department of Information Management, Chaoyang University of Technology, Taichung City, Taiwan, ROC.
4 Department of Computer Science and Information Engineering, National Taipei University, National Taipei University, New Taipei City, Taiwan, ROC.

Data availability

The data used to support the findings of this study are available from the corresponding author upon request.
U zam, M. R., Zakariyya, S. Z., Li, W., & Gelen, G. (2013). The computation of liveness enforcing supervisors from sub-models of a Petri net model of FMSs. 2013 IEEE International Conference of IEEE Region 10, TENCON 2013 (pp. 1–4). https://ieeexplore.ieee.org/document/6718803

Wang, Y.-Y., Lei, A.-F., Shen, R.-K., Yang, C.-Y., Shen, V. R. L., & Chu, Y.-H. (2019, May). Modeling and verification of an intelligent tutoring system based on Petri net theory. Mathematical Biosciences and Engineering, 16(5), 4947–4975. https://epmc.org/article/med/31499699

WoPeD. (2019, January). Workflow Petri Net Designer (WoPeD). https://woped.dhbw-karlsruhe.de/

Yang, F., Wu, N. Y., Wu, W., & Zhou, M. C. (2014, May). Petri net-based optimal one-wafer cyclic scheduling of hybrid multi-cluster tools in wafer fabrication. IEEE Transactions on Semiconductor Manufacturing, 27(2), 192–203. https://doi.org/10.1109/TSM.2014.2312192

Yang, S.-H., Lin, Y.-N., Chio gu, G.-J., Chen, M.-K., Shen, V. R. L., & Tseng, H.-Y. (2019, September). Novel shot boundary detection in news streams based on fuzzy Petri nets. Applied Artificial Intelligence, 33(12), 1035–1057. https://www.tandfonline.com/doi/full/10.1080/08839514.2019.1661118

Zhang, B., Qu, Y. Z., Ma, T., & Li, P. (2011). Optimization of consuming resource problem based on reachability graph of Petri nets. In Proceedings of the 30th Chinese Control Conference (pp. 1745–1748). http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?file name=KZLL201107003258&dbcode=CPFD&dbname= CPFD2012

Zhang, X. (2016, August). A Petri net based simulation to study the impact of customer response to stock-out on supply chain performance. Cogent Engineering, 3 (1), 1–16. https://www.cogentoa.com/article/10.1080/23311916.2016.1220112

© 2020 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license. You are free to:

Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.
You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Cogent Engineering (ISSN: 2331–1916) is published by Cogent OA, part of Taylor & Francis Group.

Publishing with Cogent OA ensures:

• Immediate, universal access to your article on publication
• High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
• Download and citation statistics for your article
• Rapid online publication
• Input from, and dialog with, expert editors and editorial boards
• Retention of full copyright of your article
• Guaranteed legacy preservation of your article
• Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com