Developed Adaptive Network Scheduling Control Algorithm that Mitigates Problems Identified in a Dedicated Scheduler Control System in Production Line

C A Okeke[1], I I Eneh [2]

[1]*Student, Department of Electrical Electronic Engineering, Enugu State University of Science and Technology Enugu, Nigeria.*

[2]*Professor, Department of Electrical Electronic Engineering, Enugu State University of Science and Technology Enugu, Nigeria.*

Corresponding Author: chukwumaokeke1@yahoo.com

Abstract: This work is on developing an adaptive network scheduling algorithm that mitigates problems identified in a dedicated scheduling control system in production line. The research was necessitated as a result of poor quality of service and reduced quantity of product due to performance degradation experienced in production line network scheduling control system of manufacturing industries. The existing dedicated network scheduler control system performance in the production line was evaluated, an adaptive network scheduler control system algorithm was developed to mitigate deficiencies found in the production line control system. The adaptive network scheduler control system algorithm developed introduced intelligence in the control system operation by enforcing priority in job scheduling and efficiently handling job queue on production line system. This maintained system stability even at increasing network load and generally improved the performance of the production line scheduling control system while maximizing cost.

Key Words: —*Adaptive Network Scheduling algorithm, Network Scheduling Control System, Production line and System Performance.*

I. INTRODUCTION

Production environment is a dynamic setting where unforeseen events can happen at any time. These disturbances will surely have an impact on the production processes. Therefore, a good scheduling system is the one that has the capability of dynamic decision making in a timely manner, while simultaneously maximizing the throughput, satisfying customer needs, and minimizing the operating costs [12]. Present age manufacturing productivity cannot stand in the absence of effective production scheduling, which has significant impact on the performance and stability of real time control systems. In real time systems, problems such as processing time delay, rush order, quality problem, unavailability of materials and machine failure are inevitable, and these brings about time lag in a dedicated scheduler control system [5].

Manuscript revised December 12, 2022; accepted December 13, 2022. Date of publication December 14, 2022.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

A longer time delay will create a network traffic that will degrade the control performance and a subsequent collapse of the entire system. As a result of these impending issues, an adaptive network scheduling control algorithm (ANSCA) was developed and implemented to efficiently prioritize time period for a production line dedicated scheduler Control System (DSCS). This promise to reduce network time delays and data packet drop out, which are the potential sources of instability in configurations embedded with network control system. [7]. The adaptive network scheduling control algorithm will improve the performance for both average and worst-case conditions of the control system, irrespective of the scheduling problems that the network control system experiences.

II. LITERATURE REVIEW

Scheduling problems exist in many manufacturing systems and the three elements which need to be mapped out are Time, Tasks and Resources. The time at which the tasks have to be performed needs to be optimized considering the available resources [3]. At the expense of limited time, limited resources and fast-growing task in the production line of any manufacturing industry, failure is inexorable. In the quest for solution Luhutyit and Pam, [6] worked on Comparative

Analysis of first come first serve (FCFS), short job next (SJN) and Round Robin (RR) Scheduling Algorithms which shows that primary role of all operating systems is job scheduling, and that a good operating system is the one that has good implementation of its job scheduling. Their work therefore compares FCFS, SJN and RR scheduling algorithms. The result clearly revealed that RR algorithm is a better scheme for a good performing operating system. But RR algorithm is restricted to processes whose Round Robin system is a preferred choice. The work of Manish and Abdul, [8] on An Improved Round Robin CPU Scheduling Algorithm, which shows that most CPU scheduling algorithms concentrates on maximizing CPU utilization, throughput and minimizing turnaround time, waiting time, response time and number of context switches for a set of requests. Among all the traditional CPU scheduling algorithms, Round Robin (RR) is the most popular but have a time-shared disadvantage.

The work of Chen *et al*., [2] on Conditional State-Based Scheduling (CSS) for networked control systems, shows that modern industrial network control systems are complicated and have dynamic workload via shared network, so the static network scheduling algorithms cannot handle dynamic workloads for lack of making on the-fly decisions. The traditional network scheduling is static and work in the NCS with predefined workload, though can guarantee timing and provide safety, but cannot handle dynamic workloads, due to lack of dynamism in their nature. So, their work centered on improving static based scheduler by implementing local schedulers on each application and allowing them to run on a timed triggered network. So that the conditional state-based scheduling inherits the deterministic property of static scheduling but has more flexibility than static scheduling in handling varying workload. However, their work cannot handle complex dynamic systems without introducing substantial delays.

The research done by Parashar and Chugh, [9] on Time Quantum based CPU Scheduling Algorithm, shows that the time quantum-based CPU scheduling algorithm is based on the traditional Round Robin (RR) criteria. The author heightened the RR efficiency by focusing on enhancing the selection of quantum time for the RR so as to mitigate average waiting time and yet minimize the number context switches that may arise due to shorter quantum. The author employs some laid down procedures, that gave a positive result, but their method is implementation specific to systems that implement Round Robin scheme only and therefore cannot guaranty effectiveness in a wider dimension scheduling.

The work of Gopi and Reddy, [4] on Mean-Difference Round Robin (MDRR) Algorithm with Dynamic Time Slice for Task Scheduling Using Real Time Applications, shows that a good algorithm optimizes CPU scheduling for real time applications. The author stated that one of the most difficult problems in designing an operating system is the timely allocation of resources to the processes and retrieving them. This problem was attended to by the review of different algorithms and proposing a new algorithm called MDRR with dynamic time quantum. The idea of MDRR with dynamic time quantum was obtained from the deficiencies found in RR. The result proves positive but the work is restricted only to systems that implement Round Robin scheme.

The research done by Phorncharoen and Sa-Nglamvibool, [10] on A Proposed Round Robin Scheduling Algorithm for

Enhancing Performance of CPU Utilization, shows that CPU scheduling is a crucial issue in computer and network operating systems, and Round Robin being the most widely used scheme has problem of appropriate time quantum. The author then proposed a RR scheduling algorithm, called DevRR, with new dynamic time quantum computed using standard deviation of burst time values and average burst time of each process in the queue. Their method gives positive result on comparing with other basic scheduling schemes. But their work has restrictions in that, if a burst time value for each process nearly approaches an average burst time, then DevRR algorithm will not be appropriate method for these data sets.

The work done by Badal *et al*., [1] on Customary Methods for CPU Scheduling shows that among variety of processes in operating system, scheduling is one of the most important decisions making practice for sharing several machine resources which persist in the memory. In this regard a number of approaches has been introduced by researcher to getting optimal scheduler. The author reviewed and characterized standard scheduling methods and found that first come first serve (FCFS) is most recommended for batch processing system like the breweries. But FCFS has trouble of large waiting time to process execution, therefore RR being a corrective measure to FCFS is then dominantly implemented in breweries producing industries. But their work being a preemptive process and using the concept of quantum time will generates context switches and slow down the system.

The research done by Srilatha *et al*., [11] on Synergy of Scheduling Algorithms Using Multiple Queues (SSAM),

shows that the performance of any processor system depends on the throughput of the central processing unit of that system. The throughput can be enhanced by reducing the average waiting time (AWT) and the amount of context switches (nCS) that piled up due to priority activities of jobs on queue waiting to run. The SSAM, focused on reducing the AWT and nCS. The results of comparing SSAM with others traditional scheme shows that it's better in CPU throughput with reduce average waiting time and number of context switches. The limitation shows that the algorithm was not considered for processes with different priorities.

III. METHOD

With an existing real life pseudo code for a production line network control system, we developed a dedicated network scheduling control system algorithm as seen in figure 1, this is to determine the system deficiencies at runtime. And then we developed an adaptive network scheduling algorithm as seen in figure 2 to mitigate the deficiencies found in the dedicated network scheduler control system and subsequently improves the performance of the production line network.

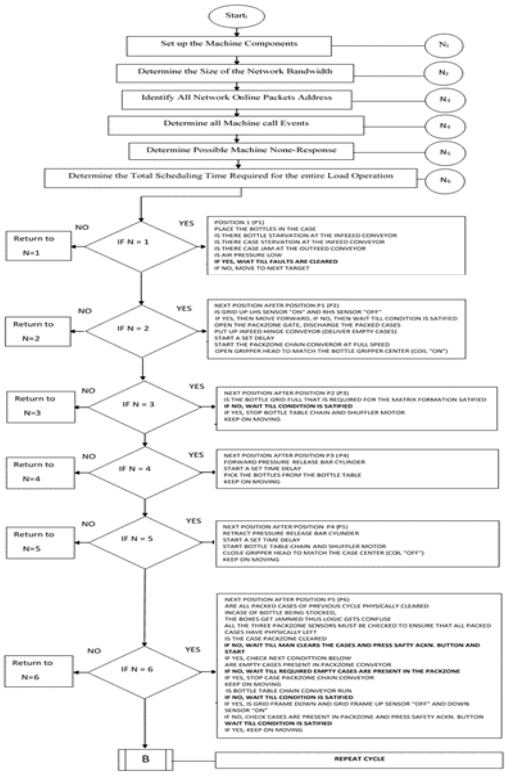

Fig.1. Flowchart for the existing network control system operation

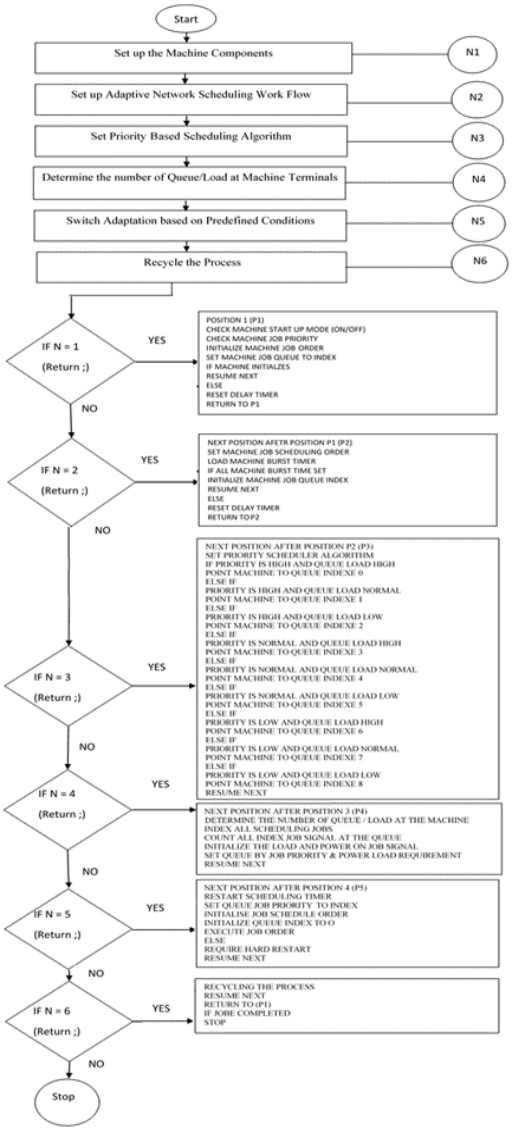

Fig.2. Flowchart for the adaptive network control system operation

IV. DEVELOPED MATHEMATICAL ALGORITHM FOR ADAPTIVE NETWORK SCHEDULING CONTROL SYSTEM

For adaptive scheduling network to perform its function at 95%, we assumed that all functions of the machine network and task event scheduling process occurs in discrete time. Considering a production line scheduler system with network delay rate (N_t) that is responsible for system downtime (y_i) and Network availability rates (N_a) being liable for system uptime (x_i). The scheduler system of the machine takes time T_i to start network scheduling priority process from state 1 to

state $1+i$ for $i \geq 0$, where the scheduling rate for state i is denoted by S_i and $i+1$ denoted by S_{i+1}. The system scheduling rate relatively depends on priority rate (P_i) of the process. When $i = 0$, the expectation rate state $[E(T_i)] = 0$, because at the initial state of scheduling $T_i = 0$. But when $i > 0$, the expectation rate state $[E(T_i)]$, can be calculated as $E(T_i) = E_{i+1}$,

where

$$E_{i+1} = \propto t_i + (1-\propto)E_i \quad (1)$$

With $\propto$ as smoothing factor and has standard values of

$0 \leq \propto \geq 1$, so that when $\propto = 0$,

$E_{i+1} = E_i$ (This history does not count) and when $\propto = 1$, $E_{i+1} = t_i$ (only the actual last scheduler processor unit burst time counts).

The scheduling performance index (SPI) can be deduced as

$$SPI = \frac{\text{Earn Value (EV)}}{\text{Planned Value (PV)}} \equiv \frac{\text{Expectected Rate } (E_i)}{\text{Priority Rate } (P_i)}, \text{ therefore}$$

$$SPI = \frac{\text{Expectected Rate } (E_i)}{\text{Priority Rate } (P_i)} \quad (2)$$

But $SV = EV - PV$, where SV is the scheduling variance which is equivalent to scheduling rate (S_i). Therefore $S_i = E_i - P_i$ (3)

Put equation 3 into 2,

$$SPI = \frac{Expectected\ Rate\ (E_i)}{Expectected\ Rate\ (E_i) - Schedulng\ Rate\ (S_i)} \quad (4)$$

Also it is given that $SPI = \frac{\text{Planned Time } (P_t)}{\text{Actual Time } (A_t)} = \frac{(P_t)}{(A_t)}$ (5)

Equation 5 into 4, we have that

$$SPI = \frac{(P_t)}{(A_t)} = \frac{(E_i)}{(E_i)-(S_i)} \quad (6)$$

But Network delay $(N_t) = P_t - A_t$,

so that $A_t = P_t - N_t$ (7)

Substitute equation (7) into (6) we have that

$$\frac{P_t}{P_t - N_t} = \frac{E_i}{E_i - S_i} \quad (8)$$

Rearranging equation 8 we have that $\frac{E_i}{E_i - S_i} = \frac{P_t}{P_t - N_t}$

$$E_i = \frac{P_t(E_i - S_i)}{P_t - N_t}$$

$$E_i = \frac{P_tE_i - P_tS_i}{P_t - N_t}$$

$$E_i = \frac{P_tE_i}{P_t - N_t} - \frac{P_tS_i}{P_t - N_t}$$

$$E_i - \frac{P_tE_i}{P_t - N_t} = -\frac{P_tS_i}{P_t - N_t}$$

$$E_i\left[1 - \frac{P_t}{P_t - N_t}\right] = -\frac{P_tS_i}{P_t - N_t}$$

$$E_i\left[1 + \frac{P_t}{N_t - P_t}\right] = \frac{P_tS_i}{N_t - P_t}$$

$$E_i\left[\frac{N_t - P_t + P_t}{N_t - P_t}\right] = \frac{P_tS_i}{N_t - P_t}$$

$$E_i\left[\frac{N_t}{N_t - P_t}\right] = \frac{P_tS_i}{N_t - P_t}$$

$$E_i = \frac{P_tS_i}{N_t - P_t}\left[\frac{N_t - P_t}{N_t}\right]$$

$$E_i = \frac{P_tS_i}{N_t} \quad (9)$$

From equation 3 and 9

$$P_i = E_i\frac{(P_t - N_t)}{P_t} \quad (10)$$

And also $S_i = \frac{E_iN_t}{P_t}$ (11)

At dynamic stage, the adaptive scheduling uses cumulative uptime occurrence rate (operational rate) x_i, cumulative downtime occurrence rate (failure rate) y_i and the total time of duration p_t.

So that Uptime rate $= \frac{x_i}{p_t}$ (12)

Downtime rate $= \frac{y_i}{p_t}$ (13)

System efficiency factor $= \frac{\text{Uptime rate}}{\text{Downtime rate}} = \frac{x_i}{y_t}$ (14)

Network availability has unavoidable impact in production line run time, and it is given as the ratio of run time to planned production time.

$$Network\ Availability = \frac{Run\ time}{Planned\ production\ time}$$

$$Network\ Availability = \frac{Uptime}{Total\ time}$$

$$Network\ Availability = \frac{Uptime}{Uptime + Downtime}$$

$$N_a = \frac{x_i}{x_i + y_i} \quad (15)$$

In dynamic scheduling process three different conditions were considered based on production.

1. Dedicated scenario (Company traditional method)
2. Adaptive scenario (Developed ANSCA)
3. Expected scenario (Company benchmark)

In condition 1, the uptime rate equal to scheduling rate, that is $(\frac{x_i}{p_t}) = (\frac{E_i N_t}{P_t})$

Obtaining the values of uptimes for six different stages and equating to 1 we have that

$$\frac{E_1N_1}{P_1}p_1 + \frac{E_2N_2}{P_2}p_2 + \frac{E_3N_3}{P_3}p_3 + \frac{E_4N_4}{P_4}p_4 + \frac{E_5N_5}{P_5}p_5 + \frac{E_6N_6}{P_6}p_6 = 1 \quad (16)$$

So, for dedicated scenario

$$N_a = \frac{[E_1N_1+E_2N_2+E_3N_3+E_4N_4+E_5N_5+E_6N_6]}{[E_1N_1+E_2N_2+E_3N_3+E_4N_4+E_5N_5+E_6N_6]+[y_1+y_2+y_3+y_4+y_5+y_6]} \quad (17)$$

Similarly in condition 2, the uptime rate equal priority rate, that is $(\frac{x_i}{p_t}) = E_i \frac{(P_t-N_t)}{P_t}$

$$\left(E_1\frac{(P_1-N_1)}{P_1}\right)p_1 + (E_2\frac{(P_2-N_2)}{P_2})p_2 + (E_3\frac{(P_3-N_3)}{P_3})p_3 + (E_4\frac{(P_4-N_4)}{P_4})p_4 + (E_5\frac{(P_5-N_5)}{P_5})p_5 + (E_6\frac{(P_6-N_6)}{P_6})p_6 = 1 \quad (18)$$

$$N_a = \frac{[E_1(P_1-N_1)+E_2(P_2-N_2)+E_3(P_3-N_3)+E_4(P_4-N_4)+E_5(P_5-N_5)+E_6(P_6-N_6)]}{[E_1(P_1-N_1)+E_2(P_2-N_2)+E_3(P_3-N_3)+E_4(P_4-N_4)+E_5(P_5-N_5)+E_6(P_6-N_6)]+[y_1+y_2+y_3+y_4+y_5+}$$

Also in condition 3, the uptime rate equal to expectation rate, that is $(\frac{x_i}{p_t}) = (\frac{P_t S_i}{N_t})$

$$\frac{P_1S_1}{N_1}P_1 + \frac{P_2S_2}{N_2}P_2 + \frac{P_3S_3}{N_3}P_3 + \frac{P_4S_4}{N_4}P_4 + \frac{P_5S_5}{N_5}P_5 + \frac{P_6S_6}{N_6}p_6 = 1 \quad (19)$$

For expectation scenario

$$N_a = \frac{\left[\frac{P_1S_1}{N_1}P_1+\frac{P_2S_2}{N_2}P_2+\frac{P_3S_3}{N_3}P_3+\frac{P_4S_4}{N_4}P_4+\frac{P_5S_5}{N_5}P_5+\frac{P_6S_6}{N_6}P_6\right]}{\left[\frac{P_1S_1}{N_1}P_1+\frac{P_2S_2}{N_2}P_2+\frac{P_3S_3}{N_3}P_3+\frac{P_4S_4}{N_4}P_4+\frac{P_5S_5}{N_5}P_5\right]+[y_1+y_2+y_3+y_4+y_5+y_6]} \quad (20)$$

These equations can initiate adaptively and predict expectation state for the adaptive scheduling control system parameter for solving overload situation and addressing resource sharing.

The algorithm for these equations is given in figure 3

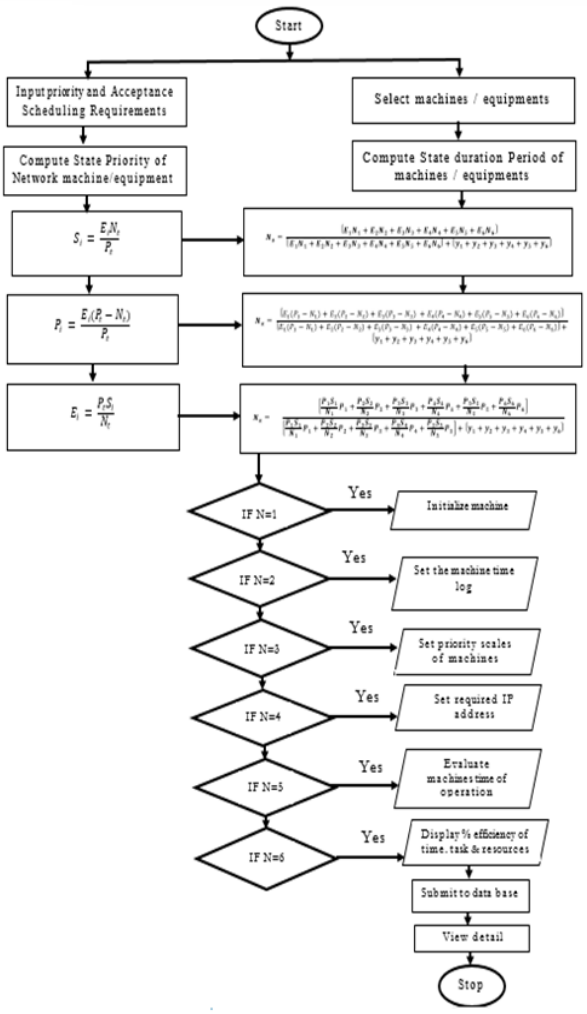

Fig.3. Flow chart of adaptive network scheduling system simulation algorithm

V. CONCLUSION

The proposed adaptive network scheduling control system algorithm was developed, mainly to prevent job queue buildup, jobs jamb and production error which arises due to job pressure on the production line of a manufacturing industries. The developed adaptive scheduling algorithm can handle real life dynamic workload as a result of its ability to make decision on the fly. This overcomes the problem of context switches, job waiting time, response time and consequently maximizes network utilization and increases the production line throughput. The adaptive network scheduling control system algorithm shows fairness in allocation of resources and enforced priority in the control system of the production line processes.

REFERENCES

[1]. Badal Dave, Surendra Yadav and Manish Mathuria, 2017. Customary Methods for CPU Scheduling: A Review. International Journal of Scientific Research in Science and Technology, Volume 3, Issue 8, Pp.344-348.

[2]. Chen Xi, Akramul Azim, Xue Liu and Sebastian Fischmeister, 2012. Conditional State-based Scheduling for Networked Control Systems. Proceeding paper in IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 19-22 August, Seoul, South Korea. Pp. 1-10.

[3]. Georgiadis Georgios P., Apostolos P. Elekidis, and Michael C. Georgiadis, 2019. Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications, Volume 7. Issue 7, Pp 1-35.

[4]. Gopi A. and Reddy Akhilesh R., 2016. Mean-Difference Round Robin (MDRR) Algorithm with Dynamic Time Slice for Task Scheduling Using Real time applications. International Journal of Engineering Research, Volume 5, Issue 11, Pp. 867-870.

[5]. John M Ikome and Grace M Kanakana, 2017. An Optimization of Manufacturing Systems using a Feedback Control Scheduling Model, Proceeding paper in International Conference on Robotics and Mechantronics. Vol 320 Pp.1-10.

[6]. Luhutyit Peter Damuut and Pam Bulus Dung, 2019. Comparative analysis of FCFS, SJN & RR Job Scheduling Algorithms. International Journal of Computer Science & Information Technology (IJCSIT) Volume 11, Issue 3, Pp.45-51.

[7]. M. Brindha and Dr J K Mendiratta, 2013. Networked Control System – A Survey, International Journal of Modern Education and Computer Science (IJMECS). Vol 6, pp. 42-48.

[8]. Manish Kumar Mishra and Abdul Kadir Khan, 2012. An Improved Round Robin CPU Scheduling algorithm, Journal of Global Research in Computer Science, Volume 3, Issue 6. Pp. 64-69.

[9]. Parashar Mayank and Chugh Amit, 2014. Time Quantum based CPU Scheduling Algorithm, International Journal of Computer Applications. Volume 98, Issue 3, Pp. 45-48.

[10]. Phorncharoen Sarayut and Sa-Nglamvibool Worawat, 2017. A Proposed Round Robin Scheduling Algorithm for Enhancing Performance of CPU Utilization, Department of Electrical Engineering, Faculty of Engineering, Mahasarakham University, Thailand, Pp. 26-28.

[11]. Srilatha N., Mounika Lakshmi Y., Naga Thulasi G., and Bharghavi K, 2018. Synergy of Scheduling Algorithms Using Multiple Queues (SSAM). Journal of Computer Engineering (IOSR-JCE), Volume 20, Issue 2, Ver. I, Pp. 63-66.

[12]. Yu Liu, Yaoguang Hu and Jingqian Wen, Yue Tang, 2018. An Overview on Smart Maintenance Service Scheduling System and Theoretical Basis for Agricultural Machinery. Paper presented in IEEE International Conference on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, 30 July – 3rd August 2018, Halifax, Canada Pp766-771.