The 6th International Conference on Manufacturing Research (ICMR08)
Brunel University, UK, 9-11th September 2008

SCHEDULING OF NON-REPETITIVE LEAN MANUFACTURING SYSTEMS UNDER UNCERTAINTY USING INTELLIGENT AGENT SIMULATION

Theopisti C. Papadopoulou *, Alireza Mousavi

Brunel University, School of Engineering & Design, Uxbridge, Middlesex, UB8 3PH, UK

* **Corresponding Author: Tel: +44 (0) 1895 265885, Email: theopisti.papadopoulou@brunel.ac.uk**

Abstract

World-class manufacturing paradigms emerge from specific types of manufacturing systems with which they remain associated until they are obsolete. Since its introduction the lean paradigm is almost exclusively implemented in repetitive manufacturing systems employing flow-shop layout configurations. Due to its inherent complexity and combinatorial nature, scheduling is one application domain whereby the implementation of manufacturing philosophies and best practices is particularly challenging. The study of the limited reported attempts to extend leanness into the scheduling of non-repetitive manufacturing systems with functional shop-floor configurations confirms that these works have adopted a similar approach which aims to transform the system mainly through reconfiguration in order to increase the degree of manufacturing repetitiveness and thus facilitate the adoption of leanness. This research proposes the use of leading edge intelligent agent simulation to extend the lean principles and techniques to the scheduling of non-repetitive production environments with functional layouts and no prior reconfiguration of any form. The simulated system is a dynamic job-shop with stochastic order arrivals and processing times operating under a variety of dispatching rules. The modelled job-shop is subject to uncertainty expressed in the form of high priority orders unexpectedly arriving at the system, order cancellations and machine breakdowns. The effect of the various forms of the stochastic disruptions considered in this study on system performance prior and post the introduction of leanness is analysed in terms of a number of time, due date and work-in-progress related performance metrics.

Keywords: Lean Manufacturing, Just-in-Time, Scheduling, Shop-Floor Control, Non-Repetitive Manufacturing, Job-Shops, Performance Modelling, Intelligent Agent Simulation

1.0 Introduction

The lean manufacturing paradigm devised by Toyota grew into a global phenomenon which is still attracting the undiminishing attention of both the industry and the academia [1]. Lean production scheduling and shop-floor control are exercised through a set of key lean concepts, techniques and tools integrated under the umbrella of Just-in-Time (JIT) pull production. Nonetheless, the majority of these critical enablers were developed in line with the design and operational characteristics of flow-shop layout configurations found in repetitive production systems in which leanness was originally introduced. This consequently led to only scarce attempts to implement the lean paradigm in the scheduling of non-repetitive manufacturing environments.

Group Technology (GT) and layout reconfigurations have been proposed in the limited attempts reported in the literature to increase the degree of manufacturing repetitiveness and facilitate the implementation of lean scheduling in complex non-repetitive production settings. Whilst the majority of these studies report satisfactory improvement in system performance resulting from the adoption of leanness they fall short to address the full size and complexity of real-life applications. More specifically they employ solution methodologies that downsize the scheduling problem considered or address a simplified version of it which often ignores the openness of the system and merely deals with its deterministic version.

Scheduling problems particularly those which are good approximations of real-life systems are highly complex combinatorial problems the optimisation of which is classified as NP-hard. The large number of input parameters, their interdependencies as well as the stochastic nature of many of these parameters calls for modelling methodologies which offer high level representation and can manage efficiently the complexity and volume of interactions pertaining ever-evolving scheduling systems. Constant advancements in computer technology coupled with the rapid evolution of simulation and artificial intelligence however, call for the issue of the transferability of leanness into the scheduling functions of complex non-repetitive manufacturing systems to be revisited. This research employs state-of-the-art agent-based simulation to extend lean pull production control to the scheduling of dynamic non-repetitive manufacturing job-shops which are subject to machine breakdowns and unexpected variations in customer demand.

The remainder of the paper is organised as follows: Section 2 presents a brief review of the literature focusing on the implementations of leanness in the scheduling of non-repetitive production systems as well as on applications of agent-based simulation in lean scheduling. Background information on job-shop scheduling and shop-floor control is presented in Section 3 along with a brief introduction to the push and pull production policies considered in the framework of this study. Section 4 gives an overview of the two agent-based architectures built to model the operation of the job-shop scheduling system under investigation and to test its performance under push and tight pull control. The section also presents the functionalities added to the agents of both architectures to model uncertainties related to unexpected demand changes and machine breakdowns. The parameters determining the experimentation setting in which the simulation runs were performed are analysed in Section 5. The simulation output from the various experimentations and comparisons drawn on the system's performance under push and the proposed lean pull shop-floor control are summarised in Section 6 which also presents brief concluding remarks on the performance of the proposed modelling methodology.

2.0 Literature Review

In spite of the general consensus in both the academia and industry that the lean paradigm is applicable merely on repetitive production systems and the subsequent lack of support to its transferability, a review of the literature reveals a small number of research works investigating the extension of leanness in the scheduling of non-repetitive production environments.

2.1 Implementations of Lean Scheduling in Non-repetitive Manufacturing Contexts

Earlier works studying the extension of leanness into non-repetitive production environments share a common point of departure. They recognise the non-repetitive nature of the manufacturing operations performed in these facilities as the strongest impediment to the application of critical lean scheduling and shop-floor control enablers. To this end, they propose functional layout adaptation or reconfiguration to increase the degree of manufacturing repetitiveness in these systems and thus facilitate the introduction of leanness into their scheduling functions. One of the first research works highlighting the need to reform job-shops into more lean-friendly shop-floor configurations is presented in [2]. The author proposes a move towards the reconfiguration of functional layouts into cellular layouts, Flexible Manufacturing Systems (FMS) or job-shop "islands". The utilization of MRP as a higher level planning and inventory management system and the implementation of JIT shop-floor control at the lower level combined with the rate per day schedules and back flushing are introduced in a reconfigured production system to support its lean transformation in [3]. Stockton and Lindley [4] propose process sequence cell layouts as an alternative to GT cells to enable the material flow to be controlled by kanbans in High Variety Low Volume (HVLV) production environments. Hybrid push/pull dual-card kanban control is implemented in different shop-floor configurations in order to study the effect of various contextual factors e.g. batch size, material handling mechanisms etc and of their trade-offs on system performance in [5].

With no prior adaptation or modification of any form to alleviate the serious restrictions imposed by certain design and operational characteristics of non-repetitive manufacturing functional configurations, early studies investigating the direct introduction of leanness into the scheduling functions of the former focused on applications not representative of the size and complexity of real-life problems. Despite their limitations these studies confirm an optimised performance resulting from the adoption of leanness. One of the first comprehensive attempts to implement leanness in a non-sequential however simplified context, is presented in [6]. In a similar study, Gravel and Price [7] employ simulation to test the performance of a job-shop under kanban control and a selection of dispatching rules developed in the framework of their work. The effects of pull control introduced in two alternative modes, i.e. tight pull and CONWIP on the performance of a Small-to-Medium Enterprise (SME) job-shop operating within a broader Make-To-Order (MTO) supply chain are modelled and analysed in [8]. A HVLV job-shop setting with stochastic arrival and processing times is considered in [9] whereby the results of the agent-based simulation showed that tight pull control exercised by kanbans outperformed the initial push system. A basestock pull control policy is introduced in a job-shop setting in [10]. No machine breakdowns or unexpected variations in demand are considered in the agent-based modelling methodology employed to test the system performance after the introduction of pull control.

2.1 Applications of Agent Based Systems in Lean Scheduling

In their majority, the applications of multi-agent systems and modelling methodologies in lean scheduling to date study the implementations of leanness in repetitive manufacturing settings utilising flow-shop layout configurations. An agent-based approach to address the problem of minimising the JIT earliness/tardiness weighted deviation in a parallel machine setting with stochastic order arrivals is proposed in [11]. An autonomous decentralised system for minimising intermediate and end product storage costs, changeover costs and due date penalties for JIT scheduling is presented in [12]. The performance of the proposed system is tested by considering a multi-stage flow-shop and experimentation results confirm the effectiveness of the proposed system in meeting the aforementioned JIT scheduling objectives while achieving considerable savings in computational time. Frey et al [13] develop a multi agent system for production planning and control which they compare with other conventional centralised approaches. The benchmarking scenario adopted in their study considers the case of a multi-level assembly where material flow is controlled by Kanbans. In a recently published study, Papadopoulou and Mousavi [14] adopt a multi-agent modelling approach to apply lean scheduling and shop-floor in a non-repetitive functional layout with particular focus on controlling the constant work-in-progress in the system. Their study considers a job-shop with dynamic order arrivals and processing times but does not account for other stochastic factors affecting the system as it assumes negligible machine downtime, order cancellations and rush order arrivals.

3.0 Lean Scheduling and Shop-floor Control in Job-shops

Job-shops are the dominant shop-floor settings in non-repetitive manufacturing environments. They employ functional layout configurations whereby equipment carrying out the same type of processing is grouped together and positioned in distinct areas of the shop-floor. Following the introduction of lean manufacturing, the two prevalent production control modes are push and pull with their names pointing to the way the system responds to actual customer demand. A job-shop operating in push mode typically comprises a number of disconnected production stages (workstations). In front of every workstation there is input buffer with theoretically infinite capacity. When actual demand information is received for a certain product type production is triggered at the first stage of its process routing. If the first station in the process sequence of this job is busy the job joins the queue of waiting jobs in the input buffer in front of the workstation. Jobs completing their processing at one workstation are pushed to the input buffer of the next workstation in the sequence without any consideration of its demand or workload.

Each production stage in a job-shop operating in pull production mode can be viewed as a production-inventory station comprising an input buffer, one or more machines and an output buffer. Apart from the movement of parts, other types of entities that move within a pull production system are demand and production authorisations. A part is released from the output buffer of a preceding stage into the input buffer of the subsequent stage in the sequence only if authorisation for the release of this specific product type is available. In contrast to the physical movement of parts downstream, the movement of customer demand takes place only logically and in the opposite direction (upstream). Production authorisations can be either physical cards (kanbans) or logical signals generated by a software scheduler.

Whilst the implementation of push production control is quite straightforward, pull production control is more complex and can be exercised by adopting various alternative pull production control policies [15]. Figure 1 below illustrates the basic principles of operation of the Kanban Control System (KCS) adopted in our study in the case of a simplified manufacturing system with two production stages in series.

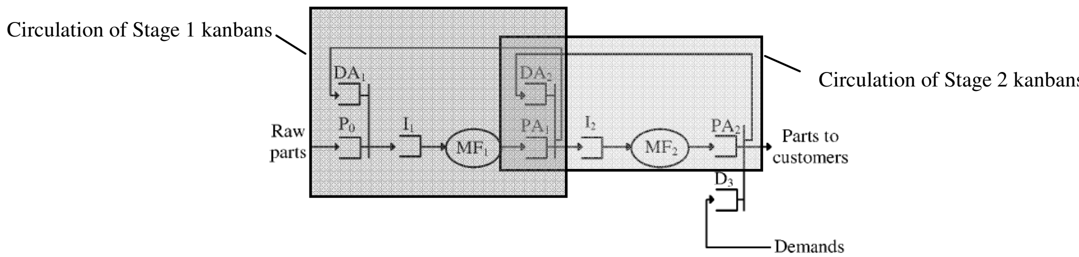

Fig. 1. KCS queuing network with synchronisation stations, case of two serial stages [16]

Queue PAi in the output buffer of stage i contains pairs of stage i processed parts and stage i production authorisations whereas queue DAi+1 denotes pairs of demand and production authorisations for the production of new stage i+1 parts. Queue Ii represents the input buffer of stage i whereas the raw material buffer and customer demand are represented as queues Po and D3 respectively.

4.0 Intelligent Agent Modelling and Simulation

The agent-based simulation models developed in the framework this study, were built using JACK Intelligent Agents™ [17]. JACK™ is a third-generation commercial framework for building and running industrial and research multi-agent applications. The framework benefits from the underlying JAVA infrastructure and multi-threading environment which offer high levels of performance, concurrency and efficiency. A multi-agent architecture is designed to model the operation of the scheduling system prior to the introduction of leanness and to benchmark its performance. This architecture is then modified to simulate the system's operation after the introduction of lean kanban-pull production control, Figure 2. Both architectures incorporate uncertainty expressed in the form of high priority orders arriving at the system unexpectedly, order cancellations and machine breakdowns.

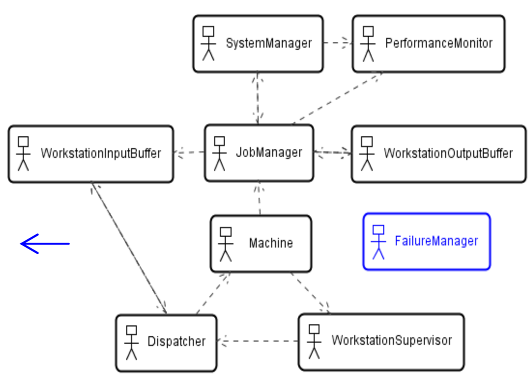

Fig. 2. Multi-agent architecture of the pull system

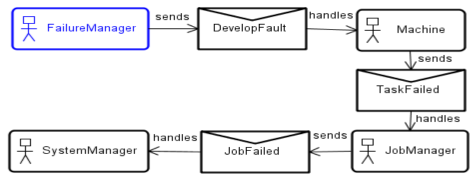

Fig. 3. Interface of the FMA

In the architecture of the initial push model the System Manager Agent (SMA) is responsible for creating the different job types processed in the modelled system. It assigns a workstation to each process step in the job's task list and sets the associated processing time. The Job Manager Agent (JMA) carries all necessary information about the job including the data determined by the SMA and other time-related data collected during its processing. The JMA manages the job's flow through the system by routing the job from one workstation (forward scheduling). When the processing of the entire job is completed, the JMA provides the job's data to the Performance Monitor Agent (PMA) which calculates the performance metrics generated in the simulation output. The input buffer queue in front of each workstation is represented by a Workstation Input Buffer Agent (WIBA). Each WIBA is responsible for exchanging information with the JMA and for updating its list following the addition/removal of jobs to/from the input buffer it manages. The Workstation Supervisor Agent (WSA) holds information on machine identification and status, i.e. busy/idle and is responsible for assigning jobs queuing in the workstation's input buffer to the machines available in the workstation. The change of machine status is communicated by the Machine Agent (MA) to the WSA whenever the machine's status changes. The last agent type available in this architecture is the Dispatcher Agent (DA) which performs the selection of the next job to be processed by employing a number of dispatching rules.

In order to model the proposed pull production system, the agent-based architecture of the initial push model is modified by introducing a new agent type, i.e. the Workstation Output Buffer Agent (WOBA). These agents exchange information with the JMA on the availability of inventory and update their databases whenever inventory is added (removed) to (from) their lists. At system initialisation their inventory lists contain pre-determined levels of zero due date inventory for all the different job types processed at the respective workstation. Further modifications to the initial model concern additional functionalities performed by the JMA. Following the arrival of a new job, its JMA requests information on the availability of a fully processed (zero due date) job from the last WOBA in the job's task list. If confirmation is received, the JMA replaces the zero due date of the already available job with the actual due date of the newly arrived job which then removes from the WOBA's database. After exchanging information with the PMA, the job agent updates the job's data by replacing its original due date with a zero due date and releases the now "zero due date" job to the system by breaking down the job's task list and executing it sequentially but in reverse order (backwards). However, if no confirmation is received, the JMA logs its request for fully processed job with the WOBA of the last workstation in the job's task list and puts the release of the job on hold until inventory is finally available.

Modelling machine breakdowns requires the introduction of the Failure Manager Agent (FMA), Figure 3. If the failure takes place whilst work is in progress without any damage caused to the part, the processing of the job will be resumed after the downtime period. However, in case of the work-in-progress being damaged, its JMA will remove the job from the machine and report back to the SMA and its life will be terminated. The cancelled job will re-enter the system and start its processing again and for that the SMA will generate a new job with a new due date to compensate for the time lost. Under pull production control a breakdown on a busy machine can only affect the zero due date replenishment jobs. A machine failure resulting in the damage of the replenishment job being processed would require the JMA to remove the affected (damaged) step of the job from the system and instigate the procedure for its replacement by a new part. In order to achieve its replacement, its JMA will simultaneously log a request for a processed part with the WOBA of all the stages in the job's process sequence preceding the stage where the breakdown occurred.

Rush orders arriving at the system unexpectedly carry a special "tag" indicating that they are high priority jobs. The functionality of the DA is modified slightly so that it initially checks whether there are any high priority jobs which it releases first before performing its prioritising functions. Under pull production control, high priority orders are filled from the available inventory immediately by treating the time of their arrival as their due date and thus as the time they need to be released to the customers. If there is no sufficient inventory to satisfy a high priority order, a request will be logged with the last WOBA in the job's sequence and will be satisfied once its inventory is replenished. A cancellation of order prior to the job's due date will result in the removal of this job from the system and the termination of the life of its JMA.

5.0 Experimentation Setting

The simulated job-shop comprises 10 machines and is processing 10 jobs with diverse process routings and number of steps between 8 and 18. The time between the arrival of jobs follows an exponential distribution with μ=0.6 hours. Processing times are generated using a uniform distribution with min=3 min and max= 10 min. The due dates are calculated using the Total Work Content Method and with the due date tightness coefficient set to 2. The dispatching rules considered are: First Come First Served (FCFS), Shortest Total Processing Time (STPT), Earliest Due Date (EDD) and Work Content in the Queue of the Next Operation (WINQ). In order to ensure the comparability of the output of the 8 simulation runs, three faults are introduced at time 5, 12 and 18 hours, with durations of 6, 4 and 5 minutes affecting the first, fifth and eighth workstation respectively. The probability damage is set to 100% for the case of workstations one and eight and 0% for workstation five. Two rush orders arrive at the system at time 10 and 15 hours and one order is cancelled at time 22 hours. The system's performance is evaluated in terms of Mean Flow Time, Mean Time in Queue, Mean Absolute Deviation (MAD) of Earliness/Tardiness, Number of Tardy Jobs and WIP.

6.0 Analysis of Simulation Output and Conclusion

In terms of the number of tardy jobs, the kanban-pull system performed better than the push system with the EDD dispatching rule producing the best results, Figure 4. However, in terms of the average number of jobs in the system at any time (WIP) the proposed kanban-pull system was significantly outperformed by the initial push system for all the dispatching rules considered and with the best of its performance observed under the WINQ rule, Figure 5. This is due the high levels of inventory maintained in the system to facilitate the operation of the kanban-pull system and achieve a satisfactory fill rate.

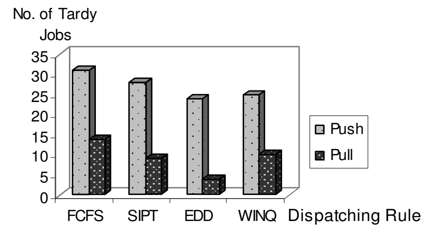

Fig. 4. Tardiness performance of push/pull systems

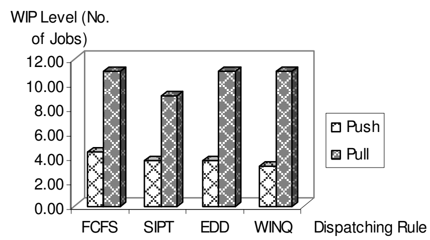

Fig. 5. WIP performance of push/pull systems

As illustrated in Figure 6, with regards to mean time in queue the best pull performance was observed under the WINQ rule whereas the EDD rule produced the least MAD of Earliness/Tardiness. Kanban-pull produced the same output in terms of mean flow time for all the dispatching rules and was outperformed by the push system which produced the best output when the WINQ rule was employed. The consistent performance of the kanban-pull system in terms of the mean flow time is attributed to the way the pull logic is implemented i.e. jobs in the available inventory are held until actual demand releases them from the system at their due date.

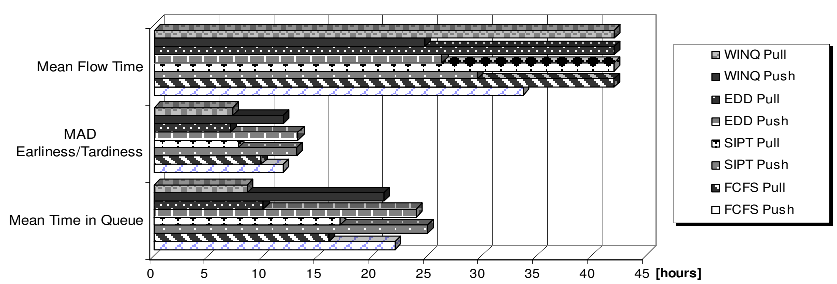

Fig. 6. Time-related performance of push/pull systems

Concluding, the employed agent-based simulation managed the complexity and stochastic nature of the scheduling system efficiently by offering high level representation and performing well in terms of computational time requirements.

References

[1] T. C. Papadopoulou and M. Ozbayrak: “Leanness: Experiences from the Journey to Date”, Journal of Manufacturing Technology Management, Vol. 16, No. 7, pp. 784-807, 2005.
[2] J. P. Kelleher: “JIT Application in the Job-shop Environment”, APICS Conference Proceedings, pp. 348-351, 1986.
[3] Sandras: “Job-shop JIT”, APICS Conference Proceedings, pp. 448-452, 1985.
[4] D. J. Stockton and R. J. Lindley: “Implementing Kanbans within High Variety/Low Volume Manufacturing Environments”, International Journal of Operations & Production Management, Vol. 15, No.7, pp.47-59, 1995.

[5] J.-W. Li and D. J. Barnes: "Investigating the Factors Influencing the Shop Performance in a Job-shop Environment with Kanban-based Production Control", International Journal of Production Research, Vol. 38, No.18, pp.4683-4699, 2000.

[6] G. A. Levasseur and R. L. Storch: "A Non-sequential Just-in-Time Simulation Model", Computers in Industrial Engineering, Vol. 30, No. 4, pp.741-752, 1996.

[7] M. Gravel and W. L. Price: "Using the Kanban Job-shop Environment", International Journal of Production Research, Vol.26, No.6, pp.1105-1118, 1998.

[8] M. Ozbayrak and T. C. Papadopoulou: "A Flexible and Adaptable Planning and Control System for an MTO Supply Chain System", Robotics and Computer-Integrated Manufacturing Vol. 22, pp.557-565, 2006.

[9] T. C. Papadopoulou, A. Mousavi and P. Broomhead: "Extending Lean Pull Production Control to the Dynamic Scheduling of HVLV Systems using Intelligent Agent Decision Support", FAIM Conference Proceedings, pp.498-505, 2007.

[10] T. C. Papadopoulou, A. Mousavi: "Performance Modelling of Dynamic Lean Job shops with Basestock Shop-floor Control using Intelligent Software Agents", ICMR Conference Proceedings, pp.185-190, 2007.

[11] A. Baccalatte, A. Gozzi, M. Paolucci, V. Queirolo and M. Tamoglia: "A Multi-agent System for Dynamic Just-in-Time Manufacturing Production Scheduling", Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2004.

[12] T. Nishi, A. Sakata, S. Hasebe and I. Hashimoto: "Autonomous Decentralised Scheduling System for Just-in-Time Production", Computer & Chemical Engineering, Vol. 24, pp.345-351, 2000.

[13] D. Frey, J. Nimis, H. Worn and P. Lockemann: "Benchmarking and Robust Multi-agent-based Production Planning and Control", Engineering Applications of Artificial Intelligence, Vol. 16, pp. 307-320, 2003.

[14] T. C. Papadopoulou and A. Mousavi: "Control of Constant Work-in-Progress in Dynamic Lean Job-shops Using a Multi-Agent System Approach", International Journal of Agile Manufacturing, Vol. 10, No. 2, pp.19-28.

[15] G. Liberopoulos, and S. Koukoumialos: "Tradeoffs between Base Stock Levels, Numbers of Kanbans and Production Lead Times in Production-inventory Systems with Advance Demand Information", International Journal of Production Economics, Vol. 96, No. 2 pp.213-232, 2005.

[16] G. Liberopoulos, and Y. Dallery: "A Unified Framework for Pull Control Mechanisms in Multi-stage Manufacturing Systems", Annals of Operations Research, Vol. 9, pp.325-355, 2000.

[17] http://www.agent-software.com/shared/products/index.html