A Comparison of Centralised and Decentralised Scheduling Methods Using a Simple Benchmark System

E. Tochev*, H. Pfifer*,
S. Ratchev*

**University of Nottingham, Nottingham, UK*
(e-mail: emil.tochev@nottingham.ac.uk)
(e-mail: harald.pfifer@nottingham.ac.uk)
(e-mail: svetan.ratchev@nottingham.ac.uk)

Abstract: This paper is intended to provide a comparison of a centralised scheduling system, a simple Multi-Agent System (MAS), and a Mixed Integer Linear Programming (MILP) formulation. The systems are tested on a simulation of a small scale flexible job shop that has machines in series and in parallel. The performance of the systems is assessed by running a batch of randomized jobs and comparing the number of late jobs and the length of time by which they are delayed. Additionally, simulations with random product failure are included to assess how well the systems perform with disruptions.

Keywords: Discrete event modelling and simulation, Multi-agent system, Scheduling, Mixed integer programming, Flexible job shop

1. INTRODUCTION

Production scheduling is a vital aspect of manufacturing, and as a result a lot of research has been done on establishing scheduling methods capable of finding optimal or near optimal build orders for specific machine and job arrangements and specifications. This paper is intended to provide a limited comparison of scheduling methods currently in use or being researched, using a simple benchmark system representing a small scale flexible job consisting of machines both in series and in parallel. The effects of product failures on the number and severity of delays are also explored.

Finding optimal solutions to many practical scheduling problems is an *NP*-hard problem (see for example Hopp & Spearman 2008; Artigues et al. 2001). Consequently, different methods have been established to perform scheduling tasks. The most common of these are outlined in the following paragraphs.

Mixed Integer Linear Programming (MILP) is a technique used to generate schedules. A variety of models and methods for creating them exist in literature, addressing both specific and generalised scenarios (see for example Manne 1960; Błażewicz et al. 1996; Unlu & Mason 2010; Li & Ierapetritou 2008; Artigues et al. 2001; Framinam et al. 2014). An objective function that requires minimisation or maximisation is constructed from known system characteristics, alongside a series of constraints. Solutions to the constraints are most often found using Branch-and-Bound (BnB) and cutting plane techniques, sometimes in combination. These can be used to generate locally optimal solutions to the optimisation problem. However, the complexity of the problem and the time required to solve it increases exponentially with the number of variable and constraints.

Priority Rule-Based Methods (PRBMs) are used to rapidly construct schedules for a system of machines (see Artigues et al. 2001; Framinam et al. 2014 for a list of examples and implementation). They are also referred to as dispatching rules and can be generally be classified as either local or global, and static or dynamic. Local rules consider the data of each task individually. Global rules consider the data of more than one task simultaneously or additional information not related to the task itself. Static rules always return the same priority index, whereas dynamic rules depend on the time at which they are calculated. A large variety of possible rules exist and have been implemented both in isolation and as a combination of rules (Otto & Otto 2014; Sabuncuoglu 1998; Artigues et al. 2001). The effectiveness of the different priority rules (and their combinations) depends on the systems that require scheduling and the criteria used to assess the results.

Multi-Agent Systems (MASs) are becoming more prevalent in manufacturing environments and are being used to determine production schedules, especially in more dynamic environments which require adjustment to changes such as machine down time, product failures or the arrival of urgent jobs (Ouelhadj & Petrovic 2009; Wong et al. 2006). The behaviours of the agents and their organisational structure determines how products travel through the system. The agents in manufacturing MASs can communicate with one another to determine the production schedule through a variety of different methods. The most common of these are negotiation protocols (Smith 1980; Krothapalli & Deshmukh 1999; Reaidy et al. 2006). Communication can be with other agents, such as broker/auctioneer agents in bidding based scheduling (Gordillo & Giret 2014; Gu et al. 1997; Wang et al. 2015), or with supervisor/mediator agents in hybrid-based MAS architectures (Maturana & Norrie 1996; Wong et al. 2006).

2. FLEXIBLE ASSEMBLY SYSTEM BENCHMARK

The manufacturing system benchmark consists of three machines in series, followed by three parallel inspection stations, as depicted in Fig. 1. It represents a simple flexible flow job shop. The premise of the benchmark is that different product types can be assembled using either all or a subset of the three machines. After the assembly, each product has to be tested in one of the inspections stations. Should the inspection fail, it is assumed that the whole product has to be redone completely. The inspection station router in Fig.1 indicates that all products are collected after assembly and assigned to one of the inspection stations. Recall that products may bypass some of the machines in the setup. However, they all have to be tested. This setup has been deemed sufficiently general for a simple production line. A large variety of machine environments have been specified in literature, including the single machine case, identical or unrelated machines in parallel, and job shops (see Artigues et al. 2001; Framinam et al. 2014; Pinedo 2016 for exhaustive lists and descriptions). All of the benchmark characteristics are based on discussions with an industrial partner.

In the benchmark, two different types of products (P1 and P2) are considered. It is assumed that P1 has to be processed by all three machines, whereas P2 only needs processing of Machine 1 (M1) and Machine 3 (M3). In a scenario where M2 is occupied but there is a P2 in its buffer that could bypass it, the product's behaviour is decided by the scheduling system in use. It can either bypass M2 or wait until M2 is available to evaluate it. In this particular system this does not need to be considered because the processing time of M1 is so much greater than that of M2 and M3 that the buffers before M2 and M3 do not fill. At the start of a benchmark simulation, a random order list of products P1 and P2 is generated. Each product on the list gets a due date (d_j) assigned. The goal of the benchmark is to assess different scheduling algorithms with respect to their capability to minimize the total number of late jobs and lateness of the production jobs.

To simplify notation, the remainder of the paper treats all stations as machines, independent of whether they are actual machines or inspection stations. Hence, Inspection Station 1 is M4, Inspection Station 2 is M5 and Inspection Station 3 is M6, respectively. Each machine and inspection station (m within the set of machines μ) has a fixed processing time (p_j^m) n time units to perform a job (j within the set of jobs J). The processing times are given in Table 1. Note that the processing times of the inspection stations p_j^4, p_j^5, and p_j^6 are not equal. The different times represent the different spatial locations of the inspections stations on the job floor and are a sum of the machine processing time and transportation time. Hence, different travel times are considered. The inspection stations determine whether a product has been successfully

Table 1. Processing times for each machine

Machine	Processing time p (time units $t.u.$)
M1	1
M2	0.2
M3	0.2
M4	1.5
M5	2.0
M6	2.5

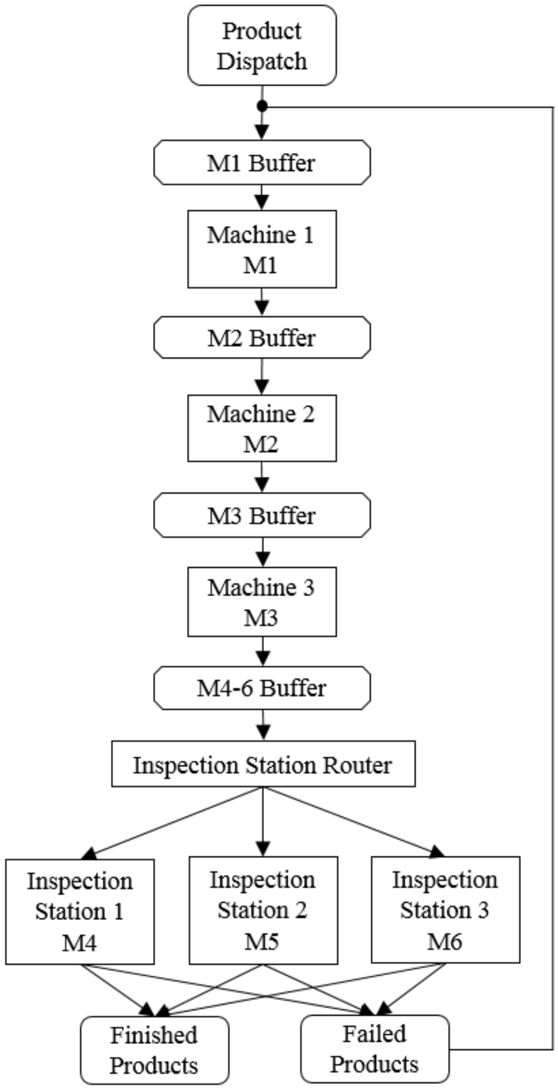

Figure 1 Machine setup in simulation

been manufactured. Completed products have their completion time (C_j) recorded and are placed in storage. It is assumed that five percent of the products fail the inspection and have to be redone. In terms of the simulation, this means that the failed product has to be added again to the order list.

A number of assumptions are made about the system to simplify the simulation, as outlined in the following paragraphs.

Transport time between stations is included in the processing times. It is outside of the scope of this paper to consider product transport mechanisms or variable transport times between machines.

A machine can process only one product at a time, and products can only be processed by one machine at a time. Machine processes cannot be interrupted. This reduces the overall model complexity. Machines 1-3 have buffers of size n_j, where n_j is the total number of products being generated to avoid product overflow. Machines 4-6 share a common buffer of size n_j because they share the processing of the entire batch of products between all three machines.

These are common assumptions and used in other pieces of research (see Artigues et al. 2001). They are justified because fully modelling all of these elements would have negligible impact on the results while increasing the complexity of the model. Additionally, this comparison is intended to analyse system performance under pressure, whereas realistically products failing inspection might be repaired outside of the main production line to avoid disrupting the schedule, or only partially reworked.

3. SCHEDULING SOLUTIONS

3.1: Priority Rule-Based Centralised Schedule

Different rules are available based on the requirements of the system. These generally use the known characteristics of the machines in the system and the products being manufactured to assign a numerical value representing the priority of the product. This is then used to establish build order by sorting the products with regards to this value.

The priority rule used in this paper will be the earliest due date (EDD) (see for example Artigues et al. 2001) and has been chosen because of its relative simplicity to implement and generally good results when compared to other PRBMs. It is also referred to as Jackson's rule (Jackson 1955), and provides optimum solutions for reducing the maximum lateness or delay in a system. EDD has drawbacks: in cases where there are products with early due dates and large processing times, these will be sequenced first despite the delays this could lead to for all following jobs. However, in this case, all products have very similar processing times so this will not be an issue.

In order to deal with product failures, the two variations of this centralised scheduling system are used. These are: a dynamic system which updates with time and the reintroduction of failed products which require reworking, and a static schedule which adds the failed products to the end of the schedule rather than adjusting it. Both of these are event driven (with the re-entry of the failed product being the event triggering the rescheduling), which is agreed to be a better rescheduling policy than periodic rescheduling by a number of studies (Ouelhadj & Petrovic 2009). Industrial scheduling uses a centralised system that often relies on periodic rescheduling, updating itself in set intervals. This is most similar to the static schedule being tested in the simulation.

3.2 Mixed Integer Linear Programming

The MILP formulation used in this paper to generate the production schedule is based on the model M3 in (Unlu & Mason 2010). This formulation is only for machines M4-6, which are in parallel, to reduce complexity, relying on Jackson's rule for an optimum product order for machines M1-3. The constraints in use are shown below. Let the binary variable $X_{jl}^m = 1$ if job j is assigned to position l on machine m; otherwise $X_{jl}^m = 0$. Additionally, variable Y_l^m is a non-negative positional date variable denoting the completion time of the job at position l on machine m.

$$\sum_{m\in M}\sum_{l\in j} X_{jl}^m = 1 \qquad j \in J \qquad (1)$$

$$\sum_{j\in J} X_{jl}^m = 1 \qquad l \in J,\ m \in \mu \qquad (2)$$

Constraints (1-2) ensures that all jobs are assigned to exactly one position on only one machine, and that each position on every machine contains at most one job respectively.

$$Y_1^m \geq \sum_{j\in J} p_j^m X_{j1}^m \qquad m \in \mu \qquad (3)$$

$$Y_l^m \geq Y_{l-1}^m + \sum_{j\in J} p_j^m X_{jl}^m \qquad m \in \mu,\ l \in J : l \geq 2 \qquad (4)$$

$$Y_l^m \geq \sum_{j\in J} (p_j^m + R_j) X_{jl}^m \qquad m \in \mu,\ l \in J \qquad (5)$$

Constraints (3-5) determine the completion time of job j at position l. Constraint (3) provides the completion time of the first job at a machine. Constraint (4) provides the completion time for the following jobs, and constraint (5) incorporates job release dates (R_j), which are determined by the completion times of jobs at M3 in the simulation.

$$C_j \geq Y_l^m - M(1 - X_{jl}^m) \qquad j \in J,\ l \in J,\ m \in \mu \qquad (6)$$

Constraint (6) provides the completion time of the product j, where M is a sufficiently large number (with a value of 100 in this case).

The cost function being minimised is as follows:

$$\min_{j\in J}(\max(C_j - d_j) + 1/n_j \sum_{j\in J} C_j) \qquad (7)$$

minimising the sum of the maximum lateness and the average completion time. The addition of the average completion time prevents the algorithm from settling on a low maximum lateness solution with a large average completion time.

3.3 MAS Solution

The MAS in this simulation consists of localised, “greedy” or self-interested sorting of products at the buffer of each machine (Wong et al. 2006). Products are locally sorted with respect to their due dates using the EDD (Earliest Due Date) rule, or Jackson’s rule (Jackson 1955). This is the same PRBM as the one being used by the centralised scheduling systems.

The choice between M4, M5 and M6, which are in parallel, is made by calculating the remaining time until each machine is available and the time it will subsequently take to complete the product whose path is being decided. The product is sent to the station offering the shortest completion time. The product is then sorted with respect to its slack while in the buffer for the inspection station.

This is a particularly simple implementation of an MAS, meant to provide a comparison against a simple centralised scheduling system. More complex MASs can include additional agents capable of coordinating the individual agent-specific schedules to produce an overarching schedule (Wong et al. 2006) or negotiating with one another (Smith 1980; Krothapalli & Deshmukh 1999; Reaidy et al. 2006) to decide product dispatch order.

4. RESULTS

The system processed a batch of 50 products, with randomly generated due dates and product types. This is intended to mimic a production line responding to small orders with short and unpredictable due dates.

The metrics most commonly used for comparison are work-in-progress and the product delays (see Artigues et al. 2001). Work-in-progress is not considered in this simulation as it is very similar to the average product delay. The metrics used for comparison are the average delay (of the delayed products), the longest delay, the total number of delays, and the total processing time of the entire batch of products. The goal of the scheduling and sorting systems is to minimize each of these metrics.

The averaged delay and number of delayed products are useful indicators of the effectiveness of a production scheduling system. The relative importance of each depends upon the costs to the manufacturer of the delayed products and the lengths for which they are delayed.

Table 2. Results of simulation with no product failures

	MAS	Centralised schedule	MILP
Average delay (*t.u.*)	3.5	2.7	2.2
Longest delay (*t.u.*)	7.4	5.4	5.2
Number of delays	49	48	43
Total processing time (*t.u.*)	53.1	51.7	51.7
Cost Function Value (*t.u.*)	35.8	33.0	32.9

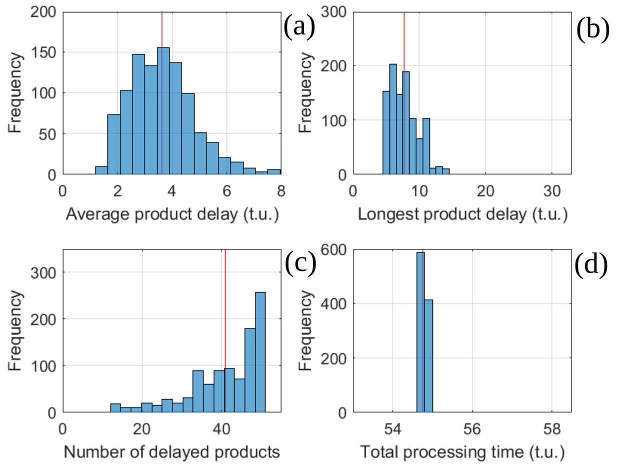

Figure 2: MAS results over 1000 simulations: (a) Average product delay, (b) Longest product delay, (c) Number of delayed products, (d) Total processing time, with averages in red

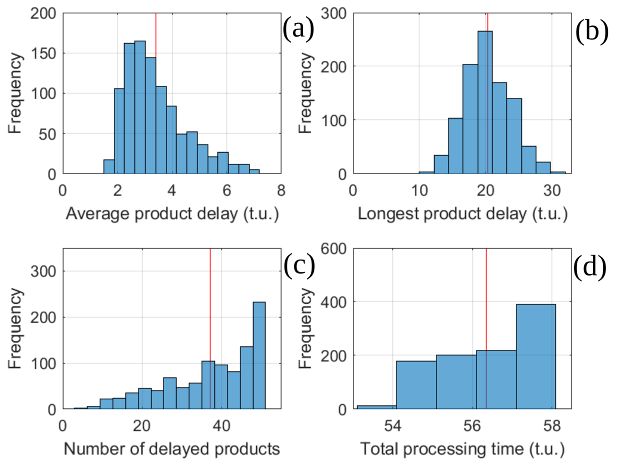

Figure 3: Static schedule results over 1000 simulations: (a) Average product delay, (b) Longest product delay, (c) Number of delayed products, (d) Total processing time, with averages in red

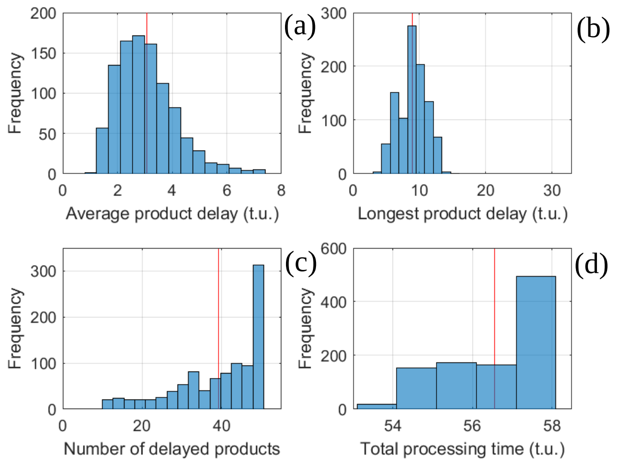

Figure 4: Dynamic schedule results over 1000 simulations: (a) Average product delay, (b) Longest product delay, (c) Number of delayed products, (d) Total processing time, with averages in red

Table 3: Average results over 1000 simulations including product failures

	MAS	Static Schedule	Dynamic Schedule
Average delay $(t.u.)$	3.6	3.4	3.1
Longest delay $(t.u.)$	7.7	20.3	8.9
Number of delays	41	37	39
Total processing time $(t.u.)$	54.8	56.3	56.6

The longest delay identifies the product with the greatest difference between its due date and its completion time. This additionally helps to provide a perspective on the distribution of delay lengths – if the longest delay is similar to the average delay, the overall variance in delay length is unlikely to be large.

The total processing time for all of the products is an indicator of the length of time required to run the production line being simulated. Minimising this value would reduce the running costs of a factory.

The results from the simulations with no failures are presented in Table 2. For this simulation, all three simulations started with identical product batches. The MILP formulation provided the best result, with a lower average delay, longest delay and number of delays than both the centralised schedule and the MAS. However, the total processing time of the centralised schedule is equal to that of the MILP. The result of the cost function (equation (7)) used by the MILP is calculated for both the MAS and centralised schedule as well, to further compare their performance. The three scheduling systems have similar values. However, the MILP has the smallest value, followed closely by the centralised system. The MAS demonstrated the worst performance of the three systems.

The simulations that included a 5% product failure rate were run 1000 times each with identical randomly generated batches of products, and the results are presented in Figures (2-4). Table 3 shows the average values of each metric, which are also presented on Figures (2-4) with red lines. The response of the MILP to product failures was not tested because of time and software constraints. The results show that for all of the systems, the average delay and number of delayed products vary over a range of values. The delays are spread over such a large range because of the random assignment of due dates for each round of simulation. These ranges are almost the same for all three systems. The longest product delays are spread over similar ranges for the MAS and dynamic schedule, with a slightly lower average value for the MAS as a result of its immediate prioritisation of any failed products passing through the system a second time. The static schedule has a wider range of longest product delays, with an average longest delay over twice the length of the other schedules. This difference occurs because of the static schedule leaving failed items to be processed last. Both the dynamic and static scheduling systems also demonstrate the same total processing times, split over the same range. The MAS has lower total processing times, split over a smaller range, indicating that it is better at responding to the random failures in this benchmark than the centralised systems do. The static and dynamic systems' total processing times are spread over the same range, but the dynamic system has a higher average total processing time.

Introducing failures to the centralised systems and MAS leads to overall worse performance, with cases where all of the products in the simulation are delayed as a result of early product failures and stringent due dates. This shows that in this benchmark system, the due dates of the initial batch of products have a large effect on the number and length of delays. This is in contrast to the total processing times for the simulations, which did not vary greatly. This indicates that the all of the systems are capable of dealing with random failure in a stable manner.

5. CONCLUSION

The systems presented in this paper represent simple implementations of common scheduling techniques. Initially, a centralised scheduling system, MILP formulation and MAS are compared using simulations running on a benchmark flexible flow job shop system. The MILP formulation creates the best schedule, followed by the centralised system. The MAS had the worst performance. The benchmark is then expanded to include product failures and dynamic and static centralised scheduling systems are compared to a MAS. In this benchmark, the MAS leads to a shorter longest delay and a smaller range of total processing times, but has a slightly larger average delay and greater number of delays.

Future work includes running simulations that incorporate product failures into the MILP formulation to assess its performance in comparison to the other systems.

6. ACKNOWLEGDEMENT

The authors would like to thank Siemens Congleton, especially Mr Carl German, for their helpful advice. This work is supported by the ESPRC iCASE entitled "Factory of the Future – Flexible, adaptable and self-optimised manufacturing systems”.

REFERENCES

Artigues, C. et al., (2001). *Production Scheduling* First Edit. P. Lopez & F. Roubellat, eds., London: ISTE.

Błażewicz, J., Domschke, W. & Pesch, E., (1996). The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research*, 93.

Framinam, J.M., Leisten, R. & García, R.R., (2014). *Manufacturing Scheduling Systems* First. Springer, ed., London: Springer-Verlag Berlin Heidelberg.

Gordillo, A. & Giret, A., (2014). Performance Evaluation of Bidding-Based Multi-Agent Scheduling Algorithms for Manufacturing Systems. *Machines*, pp.233–254.

Gu, P., Balasubramanian, S. & Norrie, D.H., (1997). Bidding-based process planning and scheduling in a multi-agent system. *Computers & Industrial Engineering*, 32(2), pp.477–496.

Hopp, W.J. & Spearman, M.L., (2008). *Factory Physics* Third Edit., New York: McGraw-Hill.

Jackson, J.R., (1955). Scheduling a production line to minimize maximum tardiness. *Research Report, Management Science Research Project*.

Krothapalli, N.K.C. & Deshmukh, a. V., (1999). Design of negotiation protocols for multi-agent manufacturing systems. *International Journal of Production Research*, 37(October), pp.1601–1624.

Li, Z. & Ierapetritou, M., (2008). Process scheduling under uncertainty: Review and challenges. *Computers and Chemical Engineering*, 32(4–5), pp.715–727.

Manne, A.S., (1960). On the Job-Shop Scheduling Problem. , 8(February 2015), pp.219–223.

Maturana, F.P. & Norrie, D.H., (1996). Multi-agent Mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing*, 7, pp.257–270.

Otto, A. & Otto, C., (2014). How to design effective priority rules: Example of simple assembly line balancing. *Computers and Industrial Engineering*, 69(1), pp.43–52.

Ouelhadj, D. & Petrovic, S., (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4), pp.417–431.

Pinedo, M.L., (2016). *Scheduling: Theory, algorithms, and systems* Fifth., Cham: Springer International Publishing.

Reaidy, J., Massotte, P. & Diep, D., (2006). Comparison of negotiation protocols in dynamic agent-based manufacturing systems. *International Journal of Production Economics*, 99(1–2), pp.117–130.

Sabuncuoglu, I., (1998). A study of scheduling rules of flexible manufacturing systems: A simulation approach. *International Journal of Production Research*, 36(2), pp.527–546.

Smith, R.G., (1980). The Contract Net Protocol: High-level communication and control in a distributed problem solver. *IEEE Tranactions on Computers*, C-29(12), pp.1104–1113.

Unlu, Y. & Mason, S.J., (2010). Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems. *Computers & Industrial Engineering*, 58(4), pp.785–800.

Wang, S. et al., (2015). Towards smart factory for Industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 101, pp.158–168.

Wong, T.N. et al., (2006). Dynamic shopfloor scheduling in multi-agent manufacturing systems. *Expert Systems with Applications*, 31(3), pp.486–494.