Multi Agent Based Control of Manufacturing Flow Shops

Duncan McFarlane[1]

Martin Kollingbaum[2]

Jeremy Matson[1]

Paul Valckenaers[2]

[1] Institute for Manufacturing

University of Cambridge

Mill Lane

Cambridge CB2 1RX, U.K.

[2] Katholieke Universiteit Leuven

Mechanical Engineering Department

Leuven, BELGIUM

Abstract: Suitable algorithms and architectures for the application of multi agent systems within a shop floor control system are developed in this paper. In particular, the algorithm discussed here is focussed on the achievement of high and consistent levels of production throughput in a manufacturing flow shop environment where there is a significant degree of flexibility associated with route selection. It is intended that in addition to performing well under nominal operating conditions, that such algorithms support a control system that is robust in the face of typical but unexpected process disturbances, such as congestion. It is emphasised that the algorithms are primarily intended for shop floor control and execution rather than scheduling for which numerous existing algorithms already exist. The paper presents a particular *back pressure* algorithm for routing selection and combines with this a simple agent environment for determining feasible routes.

Keywords: Manufacturing systems, shop floor control, multi-agent systems, back pressure, localised decision making

1. Introduction

In this paper we present results from the ESPRIT research project MASCADA [1], which investigated the application of agent technology to the distributed control of flexible flow shops – that is plants for which there are multiple, predefined production routes. *Agent-based control systems* – whose control behaviour emerges from the concerted activities of single agents - are designed in this context to be able to seamlessly manage short-term operational disruptions and longer-term changes of process/product characteristics in a "business-as-usual" manner. The agents in these control systems are chosen to represent physical objects such as orders, resources and products [2], [3], [4]. This work specifically focussed on the design of multi-agent shop floor control systems capable of a) achieving high plant throughput under disturbance situations and b) handling longer term product change, plant layout change and control software upgrades.

Central to this paper is the aspect of co-ordination and decision making in agent-based manufacturing control systems. Creation of sensible and stable control behavior within a distributed, multi-agent, software environment requires specific infrastructural mechanisms to enable interaction and information exchange between agents and special algorithms compatible with these mechanisms to manage the (distributed) decision making between the participating agents. In the context of flow shop control, work was undertaken to investigate whether it might be possible to a) define a mechanism for information exchange such that agents making local decisions could be made aware of their global implications and b) design an algorithm for local routing decision that would be compatible with this mechanism and importantly which did not require *node-specific* tuning or weighting information. That is, a single algorithm would ideally apply to multiple routing situations in different flow shops. Clearly this would greatly ease the commissioning load for the control system and also make for an algorithm that was likely to be more generalisable to other domains. An infrastructural mechanism based on the so called *pheremone* approach is described in this paper which allows the propagation of information throughout a plant topology so that such information is locally available at those decision points. Also, a class of algorithms from queuing theory - referred to as "*back pressure*" - is proposed which do not require a set of parameters to be determined for each decision point in a plant. Central also to this work is the observation that routing situations in many flow shop environments can be reduced to a small set of different topologies which greatly aids the development of simple, generic architectures and algorithms. A simplified simulation test bed based on a automobile paint plant is used to illustrate the expected level of performance of this sort of system.

2. Problem Analysis & Modelling

In this section we introduce a three part problem analysis and modelling approach and apply it too the automobile paint plant in order to extract control requirements and also to understand the extent to which simplified models can be used to meaningfully represent process behaviour.

2.1 Objective Analysis

In this phase we simply determine the main production objectives for the plant. In the case of the autombile plant, the overriding production objective is that of maintaining throughput in the face of production disruptions and to a lesser extent ensuring an acceptable residence time for individual cars. Of particular concern is the issue of conveyor blockages restricting the flow of cars to painting machines, forcing unnecessary idle time, and hence greatly reducing the throughput potential of the plant.

In summary, a clear aim of this work is to understand the type of algorithm likely to establish good throughput, with the additional requirements that the algorithm be simple, readily extendible and support reusability.

2.2 Topology and Flow Analysis

In this section we seek to examine the topology of the process under consideration in order to understand the routing issues and situations that occur. One of the difficulties in developing a control solution for a large flow shop is the complexity of the routing system and the diversity of the types of routing decision making nodes. Hence one of the key aspects of this analysis is to consider the degree to which the topology of a complex flow shop might be reduced to more basic elements to clarify the type of control architecture required and to (potentially) enable a simpler and more generalizable control algorithm to be specified.

The abstraction approach used here has been to consider the modeling of the routing processes in a typical flow shop in terms of a simple conveyor / server model where

- *servers* represent machines, switching points and inspection stations and involve time delay and/or routing decisions and can also involve quality decision which overrides routing decision making. Servers involving a time delay can also divide their time between operating and waiting (idling)
- *conveyors* - which are always one way – can have finite and small lengths and can permit only a restricted subset of products.

The advantage of such a simplified model is that from a routing perspective there are essentially only four classes of base configurations that need be deployed to construct most routing situations. (See Figure 1 where arcs represent conveyors and nodes represent servers.) Case 1 represents a pure *scheduling* situation where products items arriving on multiple conveyors must be place onto one and Case 2 represents a pure *routing* situation where a route must be selected for items on the incoming conveyor. Cases 3, 4 represent specialisations and combinations of the cases respectively.

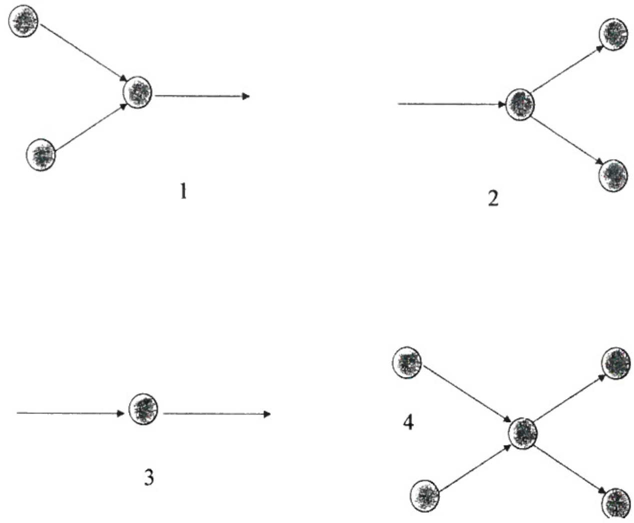


Figure 1: Basic Routing Configurations

By way of illustration, several key routing situation for the car painting plant and their equivalent conveyor/server configuration are given in Figure 2 and Figure 3. (See [5] for details.)

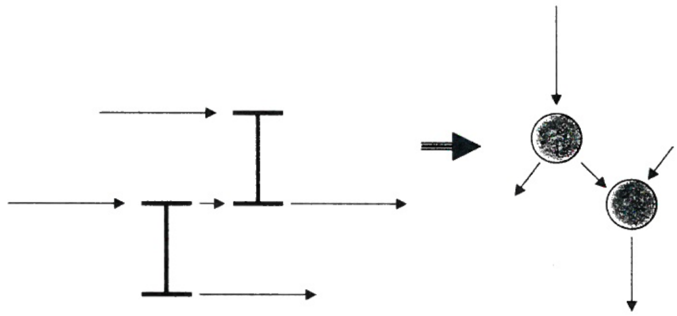


Figure 2: Conveyor/Lift Configuration

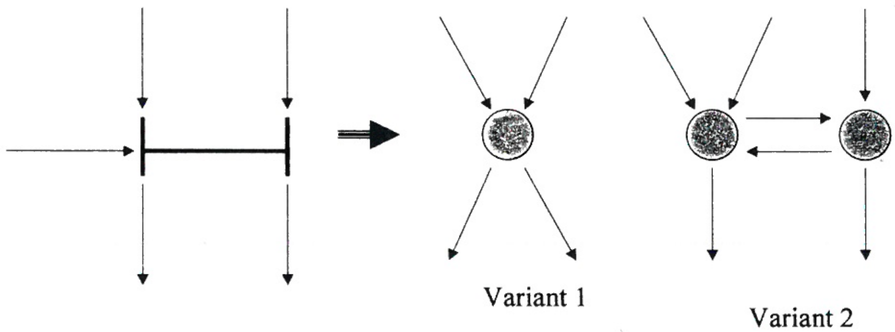


Figure 3: Lateral Conveyor with Crossover

Although some of these topological reductions represent minor simplifications of the routing behaviour of the true plant they do importantly capture the essence of the routing decisions being made in each case and hence provide an adequate model for algorithm design and analysis.

2.3 Disturbance Analysis

A key requirement for the definition of the control system specification and also for the verification of the simplified topogical modeling approach is that each adequately address the behaviour of the true plant in the face of typical disturbances. The main disturbances identified in the case of the automobile paint plant were

- yield deterioration – decrease in the yield % of a paint station caused by either paint changeover or deteriorating paint quality
- breakdown – to painting machines owing to mechanical failure or paint failure
- blocking – blocking of conveyor routing points owing to delays in downstream processing
- We are able to capture each of these within the server/conveyor framework described above:
- yield deterioration – modelled directly by an altered yield at the appropriate station
- breakdown – by reducing the overall service rate of the server in question (either switch point or paint machine)
- blocking – represented by enforced high levels of queues on particular conveyors and low service rates of subsequent servers.

3. Control System Development

In this section we outline the different stages of development of the multi agent based routing control system.

3.1 Controller Requirements

The algorithm selected for local decision making was required to be clearly aligned with the objectives and control requirements of the problem as outlined in Section 2, namely:

- Routing / Scheduling – ensuring that scheduling of multiple incoming queues (of multiple color products) and routing of products onto output queues is managed locally for all topologies
- Distributability – the algorithms should be distributable to local decision making nodes
- Maintaining Throughput – throughput must be measured and information about this measure be made available in all routing decision making.
- Reducing Blocking – the algorithm needs to be able to sense current/future queues of products
- Multiple product types – the algorithm must apply to non homogenous product environments

An additional requirement observed to be beneficial on the shop floor is that of batching the cars into groups of the same color. The development of the agent based control system is divided into two parts: a) The development of a feasible route selection mechanism – a means for determining at a given node which routing options are feasible given the production requirements for the product in question and b) The development of a best or "optimal" route selector which enables differentiation between different feasible routing possibilities.

3.2 Feasible Route Selection

In the following we will outline how information distribution for decision making by single resources is enabled in this environment, and hence how feasible routes can be clearly signposted at decision nodes to oncoming products. Streams of products must be guided over a network of transport facilities and routing points towards processing resources. These streams of product entities must be guided in a way such that a) they only follow permitted routes and b) a high utilization of processing facilities is made possible. Entities stream over a sequence of nodes or decision points on their way towards the processing resources. At such points, information must be available, which supports routing decisions made by product or resource agents. For this purpose, global information is made accessible locally at a specific decision point via an information distribution mechanism. For that purpose we have to provide both *plant information* (representing the network of conveyors and servers) to support this propagation mechanism, and product and *product information* to allow the definition of sequences of processing steps, which a product has to undergo at different processing units. This capability is represented through a mechanism referred to as a *subnet.* When a product is transported towards a processing resource, which can perform the next required processing step, it will (maybe) be routed over a sequence of decision nodes. Each decision node will have one or more outgoing conveyors as principal routing options and a set of incoming conveyors, from where products should be forwarded to one of the output conveyors. Only a sub-set of these output conveyors will lead into the section of the plant where machines which can offer the next processing step are located. This information is made available at the local decision making nodes by propagating processing capability information upstream.

Figure 4 illustrates subnets of different capabilities for a number of different nodes.

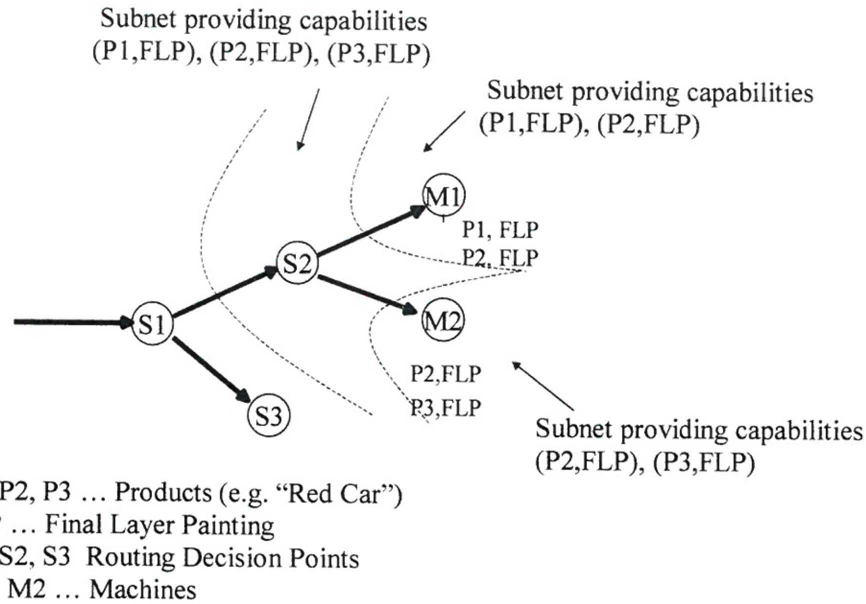


Figure 4: Subnet Processing Capabilities

A decision node (routing agent) is informed about the capabilities of each sub-net and can – based on this information – make an informed routing decision across all feasible routes by applying a specific algorithm which we discuss next.

3.3 "Optimising" Route Selection

Route selection involves determining which incoming conveyor to select products from (scheduling) and which outgoing conveyor to dispatch the selected product onto (routing). A number of different candidate algorithms were considered as a means of addressing the problem defined by the analysis of the car painting plant and its production objectives. The algorithm class chosen as most suitable for the purposes of this work was the so called *back pressure reduction* algorithm as a means of handling the different routing and local "scheduling" decisions for the topologies mentioned above . Back pressure methods originated in telecommunications network control [6] and are designed specifically for congestion management. These algorithms are able to handle arbitrary topologies, multiple traffic types and – importantly – systems where looping of products is common. Hence they meet most of the requirements outlined in the previous section. Additionally, there are demonstrated connections with maintaining throughput levels for a particular sub class of systems [6]. The back pressure algorithm is illustrated in its simplest form next. illustrates a server with cars of multiple color types arriving and leaving for downstream processing. In the following analysis we ignore the different colors and simply consider aggregate flows but note that an equivalent multiple color version has been developed for this work.

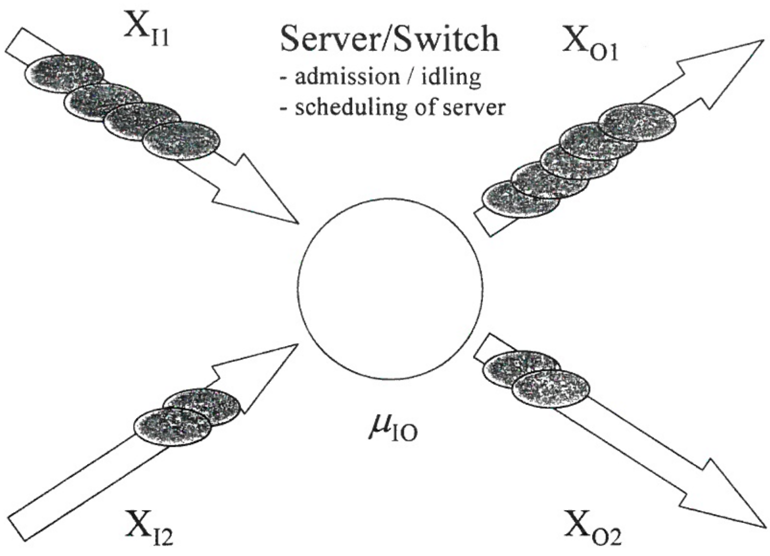


Figure 5: Back Pressure Nomenclature

Let (I^c, O^c) be the currently selected (I, O) pair and define

$$A^c(t) = \mu_{I^c O^c}\{X_{I^c}(t) - X_{O^c}(t)\}$$

where X describes the queue length at input and output conveyors, □ the service rate of the switch node, and A^c the current back pressure for the current I/O pair. The switch node – which is the decision making element – continually calculate the following quantity based on the set of all Input/Output pairs:

$$A^*(t) = \max_I \{\mu_{IO} \max_O \{X_I(t) - X_O(t)\}\}$$

Then, at each time or event update at that server, calculate both $A^c(t)$ and $A^*(t)$.

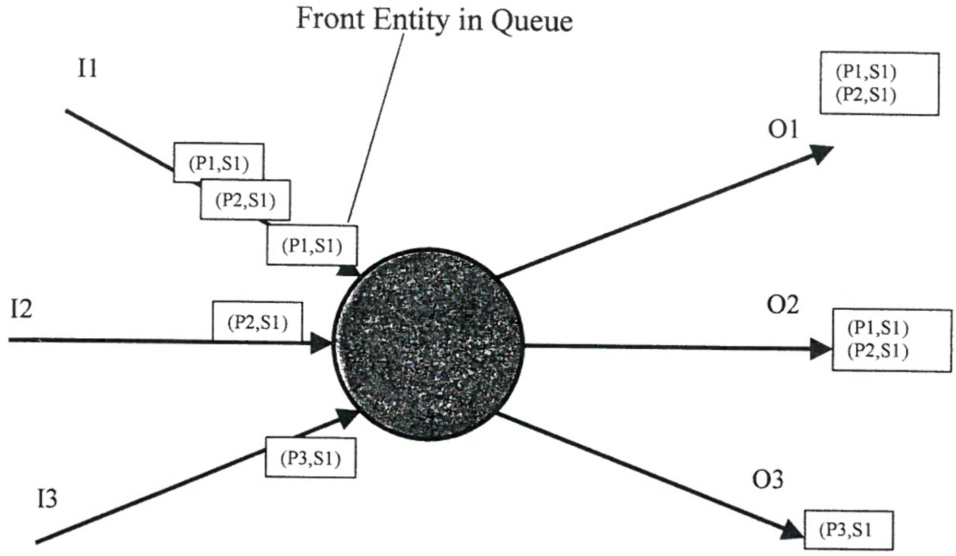
If $A^*(t) \leq 0$,
then the server idles.
If $A^*(t) > 0$ and $A^*(t) \leq \alpha\ A^c(t)$,
then continue routing via (I^c, O^c).
If $A^*(t) > 0$ and $A^*(t) > \alpha\ A^c(t)$,
then switch to $I^c = I^*$ and $O^c = O^*$.

Here $\alpha > 1$ is a tunable parameter which prevents rapid oscillations between I/O routes and which in practice reflects the fact that changing I/O pairings can involve a time penalty. Hence at any given time point, the back pressure algorithm can be seen as selecting the highest entry from a matrix whose elements are governed by input/output pairs.

3.4 Implementing Multi Agent Control

We now describe the application of the back pressure algorithm to the subnet capability information distribution environment. The back pressure algorithm as described in Section 3.3 is limited in that it assumes only one product type and unlimited queues. In the case of the plant topologies and product types we are examining, these assumptions are not feasible, and the back pressure algorithm must therefore be adapted accordingly. We have to deal with finite conveyor length and a mix of different products on the queues. These products can only be forwarded to the output conveyors which lead into a sub-net that offers the right capabilities. The next product to be forwarded is selected from among the front products for each conveyor. Each such product will drive the search for sets of capable output conveyors. When all capable output conveyors for all input conveyors are found, a decision making algorithm can be applied on this information. The process of making a routing decision is started by creating the so-called "Pressure Table". (Refer to Figure 6.) All currently possible input-output combinations are collected into this table. The criteria, if an input-output conveyor pair has to be added to this table, are the following:

- the input conveyor is not empty
- the "front product" on the input queue requires a capability, which matches with a pheromone residing at the output conveyor



Pressure Table (capable outputs):

Input	Output	QueuePressure	FrontElement
I1	O1	10	(P1,S1)
I1	O2	2	(P1,S1)
I2	O1	8	(P2,S1)
I2	O2	7	(P2,S1)
I3	O3	1	(P3,S1)

Select highest pressure (I,O) pair in Pressure Table

Input	Output	QueuePressure	FrontElement
I1	O1	10	(P1,S1)

Figure 6: Back Pressure handling Multiple Products

In the next step a decision algorithm uses this table to find an input-output pair. If we use e.g. random choice, a pair would be randomly chosen from this table. In using back pressure, a pressure value must be calculated for each table entry. The highest pressure determines the input-output pair to choose. In case of a tie, an arbitrary choice decides between table entries of equal pressure.

3.5 Extensions to the Routing Control System Implementation

Before going on to discuss results, in this section we briefly outline some extensions to the basic back pressure algorithm explored in this study.

3.4.1 Handling Finite Conveyors and Multiple Product Types

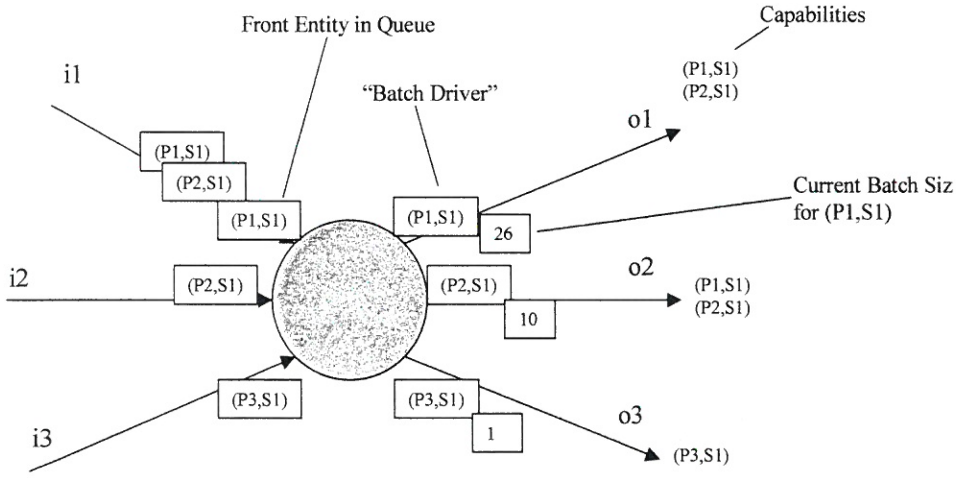
Tassiulas [6] originally uses queue lengths at nodes in a telecommunication network. His approach assumes unlimited queues, which is not appropriate in case of a manufacturing environment. Therefore different measures have to be taken. We have investigated back pressure in following different modes:

- "normal" back pressure: This form of back pressure is very near to the traditional approach, except for the handling of multiple product types; queues are assumed to be unlimited; this approach does only make sense, if all input and output conveyors at a decision node have equal length
- "relative" back pressure: at each conveyor the current percentage of occupation is calculated and used in the pressure calculation; this type of back pressure takes conveyor-length into account, when calculating a pressure

- "complementary" back pressure: in this approach we use "places available" rather then "places filled" to calculate a pressure; this approach will lead to problems if conveyor-lengths are different from each other
- "total" back pressure: if a conveyor is full, the pressure calculation extends into the sub-nets and takes also pressure on conveyors further upstream/downstream into account; with this approach we reintroduce a kind of unlimited queues and can apply normal back pressure

3.4.2 Batching of Products

A batching capability (Figure 7) can also be introduced by extending a decision node with the appropriate functionality. Building ~~batches means~~ to supply one output conveyor with products of identical type over a longer period. This is done by further narrowing down the set of possible output conveyors by fixing an output conveyor to a specific product type and extending the pressure table with a batch pressure. We introduce a so-called "Batch Driver" on output conveyors, which determine the type of product which should be forwarded to this output conveyor. Each output conveyor is also annotated with a batch counter to determine when a batch can break (long batches are made more likely to break).



Pressure Table

Input	Output	QueuePressure	BatchPressure	FrontElement	BatchDriver
I1	O1	10	26	(P1,S1)	(P1,S1)
I1	O2	2			
I2	O1	8			
I2	O2	7	10	(P2,S1)	(P2,S1)
I3	O3	1	1	(P3,S1)	(P3,S1)

Figure 7: Batching Capability

The batching mechanism has following functionality:

- When no batch drivers are set, the back pressure algorithm kicks in and determines an input-output pair and sets a new batch driver
- If no entry in the pressure table has a "Front Element" that matches with a "Batch Driver," then again back pressure kicks in to determine an input-output pair
- if a match between a "Front Element" and a "Batch Driver" is found, then a special form of pressure is calculated according to following formula:

*pressure value(in,out) = queue pressure(in,out) - □ * batch pressure(out)*

If there is no batch, normal queue pressure is relevant and decisions are taken according to back pressure. If there is a batch, it will influence the pressure value and the decision made, where □ is a tuning parameter.

4. Evaluation of Control System

The objectives of this section are to provide an evaluation of the software environment and control algorithms developed. This evaluation is carried out using a simple example topology related to the automobile paint plant we have discussed previously. We will only discuss short term disturbance management here.

4.1 Testing Environment

A test case was created, using building blocks from a modular modeling, simulation and control design framework described in [5] to construct a simplified plant topology related to the car paint shop as shown in Figure 8.

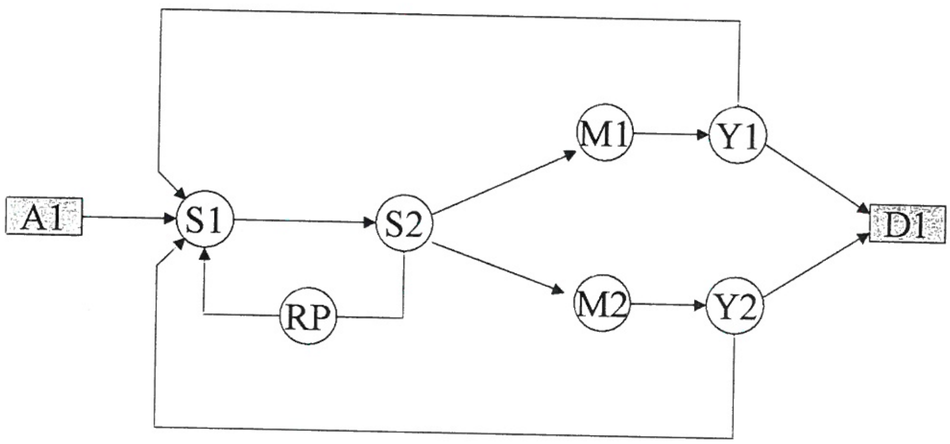


Figure 8: Topology Based on Car Painting Plant (where *M* denotes machines, *S* switch points, *Y* inspection stations, *RP* repair stations.)

An evaluation was carried out based on the topology in

Figure 8 with model parameters as described below. In the particular instance of this topology considered here, 5 different products may be produced. Machines M1 and M2 are each capable of producing 3 products, one of which is producable on both. This topology was chosen (see [3]) because it reflects a number of important complexities present in the paintshop testbed such as:

- The conveyor joining S1 and S2 carries cars with different product goals (any one of 5) and with different processing requirements (RP or FLP).
- Cars which have completed the rework/repair preparation must be fed back via S1 into the main stream of cars entering the plant.
- Machines M1 and M2 are fed via a single conveyor, entering S2 where products are routed to a machine capable of carrying out their required next production step. Some products may only be processed on one of the two machines.
- The possibility of processing machine breakdowns.
- The possibility of major yield reductions being detected at checking stations.
- The need to balance the loading/utilisation of the machines.

4.2 Performance Assessment

The above detailed topological description includes details of process dynamics, variabilities and disturbances, which we now draw on to help make a prediction, via simulation of the operational performance of the candidate algorithms. Two distinct algorithms were tested for scheduling (input selection) and routing(output selection) decisions at the servers S1 and S2. The first of these algorithms was the back pressure algorithm, the second algorithm was based on a random selection from the set of feasible input/output conveyor pairs (as determined by the subnet capability pheromones). The primary performance measure in this plant is the number of entities leaving the system per unit time of operation. (i.e. throughput). The following two assessment conditions have been addressed for the simple topology:

A :Normal Operating Disturbances with varying arrival rates where normal operating disturbances include production mix, arrival time and service time variations.

B: Normal Operating Disturbances as above together with a machine disturbance with a prescribed occurrence probability distribution and duration distribution.

4.3 Results

Experiments were carried out for the assessment conditions A and B described in the previous section. The outcome of the experiments described above is now summarised. In
Figure 9 for the case of normal operating disturbances, the back pressure algorithm provides a substantial improvement in average hourly throughput performance compared to random selection. (Note that the theoretical maximum throughput under the normal operating conditions is 19.45 cars per hour.) In Figure 10 the benefits of the backpressure algorithm in the case of machine breakdown disturbances are less pronounced as for this simple topology, breakdown reduces decision making to a minimal level. (The theoretical maximum throughput under breakdown conditions in 16.67 car per hour.) Figure 11, and Figure 12 examine the utilisation of the two machines under different arrival rates. Under back-pressure control, the production system is clearly more able to convert the load offered to the system into throughput than its random selection counterpart.

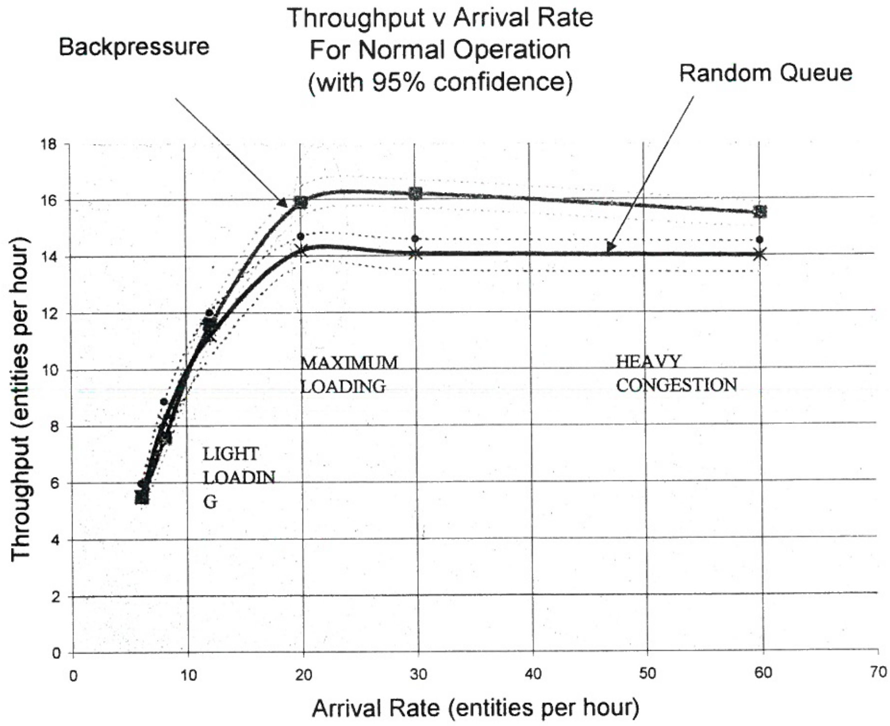


Figure 9: Throughput under Normal Operating Conditions (including +/- 5% confidence levels)

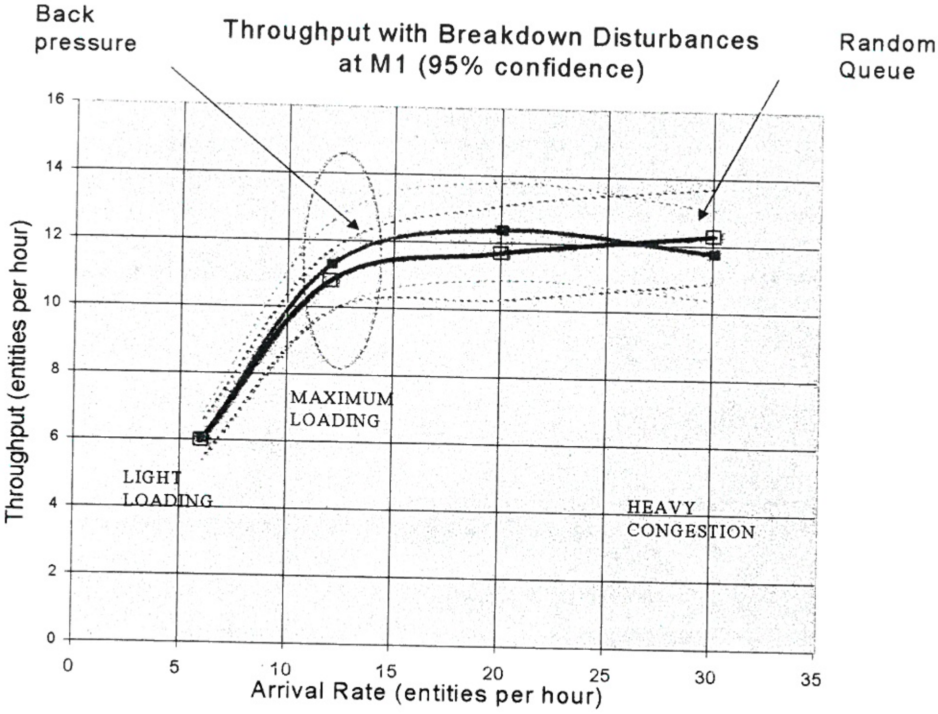


Figure 10: Throughput in the presence of machine breakdowns on Machine M1

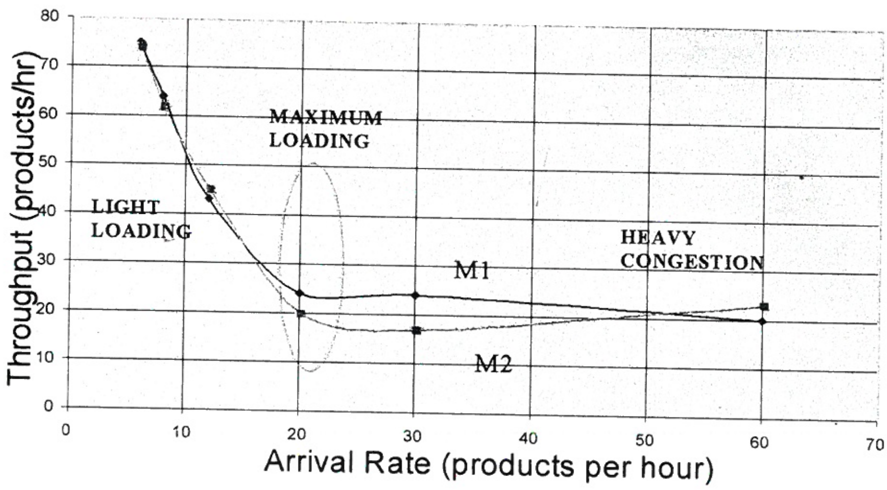


Figure 11: Percentage Idle Times for Machines M1 and M2 with backpressure

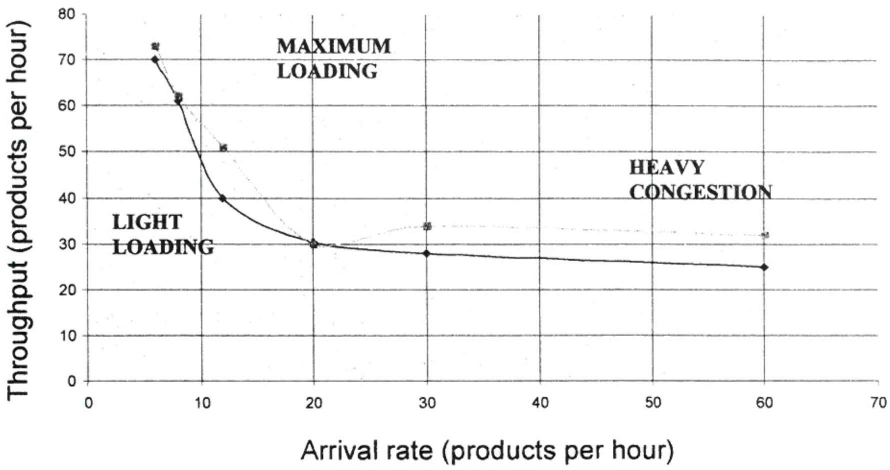


Figure 12: Percentage Idle Times for Machines M1 and M2 with random control

REFERENCES

1. MASCADA, **Management and Control of Manufacturing Systems in the Face of Disturbance and Change - An Application of Co-Operative Agents**, in http://mech.kul.ac.be/pma/projects/mascada. 1998.
2. PEETERS, P., et al. **Pheromone-based Emergent Shop Floor Control Systems for flexible Flow Shops**. in IWES'99. 1999. Kobe, Japan.
3. KOLLINGBAUM, M., et al. **Emergent Flow Shop Control Based On Mascada Agents**. in MIM 2000. 2000. Patras, Greece.
4. BRUCKNER, S., **Return from the Ant: Synthetic Ecosystems for Manufacturing Control**, 2000, Humboldt Universitat: Berlin.
5. MCFARLANE, D et al, **Development of Agent Based Control of Manufacturing Flow Shops**, in Proc of IEEE SMC 2001, Tucson, Arizona, 2001
6. TASSIULAS, T., **Adaptive Back-Pressure Congestion Control Based on Local Information**. IEEE Transactions on Automatic Control, 1995. 40(2): p. 236-250.