Distributed Scheduling of an Assembly Shop

A. Tharumarajah*, R. Bemelman*, P. Welgama** and A. Wells*
* CSIRO Manufacturing Science and Technology, Woodville, South Australia
** CSIRO Centre for Mathematical and Information Sciences, Glen Osmond, South Australia

ABSTRACT

Past approaches to solving the assembly scheduling problem have mainly been from the consideration of a hierarchically controlled shop. Thus, the problem is considered in its entirety and solved in a monolithic fashion. However, in recent times, there have been attempts to impose heterarchical or distributed control. This paper extends our previous work in the area of distributed behaviour-based control and examines the performance of a model based on this for scheduling assembly problems. Using this approach, we investigate the performance of a set of scheduling problems for a 3-stage 4-station assembly problem. This performance is compared with a Mixed Integer Programming (MIP) model of the same situation. The results are promising and show comparative due date and makespan performance.

1. INTRODUCTION

A typical assembly scheduling problem is characterised by the presence of converging material flows and precedence dependency. This invariably gives rise to presence of operations that may be performed concurrently and whose coordination at an assembly station may be critical to achieve the desired performance. For instance, when such operations are not synchronised it may lead to increases in wait time for the components and may contribute to starvation and blocking tendencies for operations up or down stream to the assembly station.

The past approaches to solving the assembly scheduling problem have mainly been from the consideration of a hierarchically controlled shop. Thus, the problem is considered in its entirety and solved in a monolithic fashion. However, in recent times, there have been attempts to impose heterarchical or distributed control [2]. In part, this is to manage the complexity of the problem, while attempting to improve the modularity and come up with controllers that can be independently configured at the level of the individual station.

The previous version of our distributed behaviour-based control model has been tested for job-shop piece-part type scheduling problems and has been found to produce makespan and tardiness performance comparable to well-known heuristics [11]. In this paper we extend our previous work and examine the performance of the Behaviour-based control model for scheduling assembly problems. In the approach we use, the first step is to distribute the scheduling problem to the individual stations. The stations then schedule their operations independently to maintain their own local schedules for the tasks performed. They may engage in communication with each other to resolve conflicts among their local schedules and coordinate the flow of components. Using this approach, we investigate the performance of a set of scheduling problems for a 3-stage 4-station assembly situation. This performance is compared with a Mixed Integer Programming (MIP) model of the same situation which endeavours to find exact or optimal solutions.

In the next section, we review the past approaches to assembly scheduling and attempts at solving it as a distributed problem. Following this, we formulate the MIP model for the situation in hand. Section 4 outlines the features of the distributed scheduling approach. The design of the experiment and results of testing the model for a set of problems are given in Sections 5 and 6. Section 6 includes comparison of the performance of the model with the corresponding MIP model. Finally, conclusions are drawn.

2. REVIEW OF PAST APPROACHES

Although many research works have been reported in job-shop and flow-shop scheduling problems, extremely few have considered the assembly type environment. Lee *et al.* [5] have considered a 3 work station assembly type flow-shop problem where they focussed on minimising makespan. Agnetis *et al.* [1] have reported a good practical application in an automobile factory. They have considered a multi-stage multi-processor assembly area. Three types of rules, push, pull and matching, have been tested using simulation. A printed circuit board assembly process has been considered by Kim *et al.* [3].

However, the above studies implicitly assumed a hierarchical control situation where the assembly scheduling problem was solved as a whole. In contrast, a typical heterarchical control situation may demand a decomposition and distribution of the sub-problems and their solutions to the various entities on the shop-floor. Such an approach has been attempted by Gou et al. [2] where they consider three different robotic assembly stations served by automated guided vehicles (AGVs). The

0-7803-4778-1/98 $10.00 © 1998 IEEE

assembly problem is decomposed and distributed to both products (as active scheduling entities) that are to be assembled and the stations (the AGV's are not modelled explicitly). These entities are considered to be acting autonomously and are called holons. A product holon manages the precedence conflicts among the operations that are required to make it, while the station holons manage the capacity allocation to perform the operations. Thus, a product holon may request operation beginning times from the station holons, and the station holons solve their own sub-problems of capacity allocation and reply with feasible time slots. Gou et al.[ibid] use a modified version of Lagrangian relaxation (LR) to solve the sub-problems and the overall problem. Thus, a part holon may compute its precedence prices (i.e., the LR multipliers) in lieu of the capacity prices (i.e., for the start times) submitted by the station holons. In turn, a station holon may compute its capacity prices from the requirements of the operation as submitted by the part holons. This represents the coordination framework among the parts and the stations. To resolve any conflicts arising from these actions, a coordinator uses the price information as a starting point to further adjust the prices and to iteratively improve the overall problem. It may happen that all the conflicts are not resolved due to limitations in computing time for real-time operation. These conflicts may be resolved through rescheduling at a later time or as conditions change. A similar approach of using the LR method is reported in Ramaswamy and Joshi [8]. Here, a part directly negotiates with the machines for awarding operations through an auction model.

Other approaches using concepts of distributed scheduling have been reported, though not for assembly problems. A survey of these approaches can be found in Tharumarajah and Bemelman [10]. The approach we consider for solving the assembly scheduling problem is quite different from all of these. It uses a heterarchical model without a coordinator to resolve conflicts. The other major difference is the feature that is implemented by each distributed entity of resolving conflicts by a process of adaptation. Section 4 highlights the features of this approach.

3. MIP FORMULATION OF THE ASSEMBLY SCHEDULING PROBLEM

Consider a simple three stage four station assembly shop as depicted in Figure 1. The shop assembles a number of products which follow the same sequence of operations. First, the stations WC1 and WC2 produce the needed components. These components (one component each from WC1 and WC2 respectively) are assembled at station WC3. This operation is followed by packaging at WC4. We consider the problem of scheduling orders (jobs) through this four station assembly shop. The criterion used for evaluating different schedules is the total tardiness. Tardiness is important in practice as most jobs have due dates and the schedulers are interested in meeting these due dates.

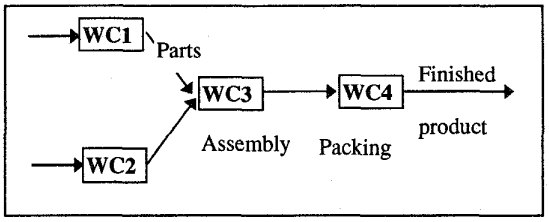

Figure 1: 3-Stage 4-Station Assembly Shop

Notation:

d_{ik} - processing time of order (job) i on machine k

i, j - indices for orders

k - index for machines (Work centers) k

m - number of machines

$x_{i.k}$ - starting time of order (job) i on machine k

y_{ij} - $\begin{cases} 1 & \text{if order } i \text{ precedes order } j \\ 0 & \text{otherwise} \end{cases}$

L_i - due date of order i

N - number of Jobs

M - a large number

Z_i^+ - tardiness of order i

Z_i^- - earliness of order i

$$\text{Minimise } Z = \left[\sum_{i=1}^{N} Z_i^+ \right] \tag{1}$$

Subject to:

$$x_{i3} \geq x_{i1} + d_{i1} \quad \forall\ i \tag{2}$$

$$x_{i3} \geq x_{i2} + d_{i2} \quad \forall\ i \tag{3}$$

$$x_{i4} \geq x_{i3} + d_{i3} \quad \forall\ i \tag{4}$$

$$x_{i4} + d_{i4} - L_i = Z_i^+ - Z_i^- \quad \forall\ i \tag{5}$$

$$x_{ik} - x_{jk} + M * y_{ij} \geq d_{jk} \quad \forall\ (j<i), k \tag{6}$$

$$x_{jk} - x_{ik} + M * (1 - y_{ij}) \geq d_{ik} \quad \forall\ (j<i), k \tag{7}$$

$$x_{ik} \geq 0 \quad \forall\ i,k \tag{8}$$

$$Z_i^+ \geq 0 \quad \forall\ i \tag{9}$$

$$Z_i^- \geq 0 \quad \forall\ i \tag{10}$$

$$y_{ij} = (0,1) \quad \forall\ (j<i) \tag{11}$$

The MIP model used here is adapted from a job-shop model proposed in [7]. Computational performance of the following model was improved by adding some model tightening constraints, and more details of these can be found in [12].

This model assumes a permutation schedule which means that the same sequence of jobs will be used for loading at each machine. This contrasts with the situation where the sequence of jobs at each machine is different. Equation (1) is the objective function to minimise tardiness. Equations (2), (3) and (4) represent precedence constraints for all orders. Equation (2) enforces the processing on WC 1 before WC 3. Equation (3) enforces the processing on WC 2 before WC 3. Similarly, equation (4) enforces the processing on WC 3 before WC 4. Equation (5) defines tardiness and earliness. Equations (7) and (8) show the machine interference constraints, that is for each machine (k), either order (i) should precede (j) or (j) should precede (i).

4. DISTRIBUTED SCHEDULING MODEL

The approach we take can be described by considering each station in Figure 1 as acting autonomously in making scheduling decisions. Initially, the stations are each assigned a set of operations in accordance with the process plans and have an estimation, for each operation, of lower and upper time boundaries (called artificial demand windows or ADWs). However, later revisions may be made by the processing stations due to the difficulty in allocating an operation. Invariably, such revisions affect the time limits of the ADWs of adjacent operations performed by other stations, and this may lead to conflicts.

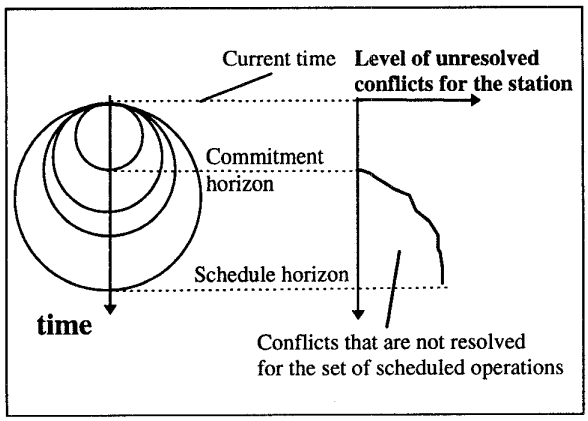

Figure 2: Impact of commitment horizon on the level of unresolved conflicts

When a conflict arises a station may communicate it to the stations processing the related operations. For instance, a violation of the boundaries of an operation processed by station WC3 (see Figure 1), may be communicated to both the up stream (WC1 and WC2) and down stream (WC4) stations. On receiving such conflicts, these stations may resort to rescheduling and may, in turn, produce conflicts and so on. However, such conflict propagation and acts of rescheduling can become unmanageable and may lead to ripple effects. This is a particularly difficult problem to overcome in distributed scheduling. Most of the approaches for dealing with this suggest negotiations (see [4] for example). Nevertheless, even with negotiations there may be situations where an agreement on resolving a conflict between any two related operations by the respective scheduling entities may give rise to secondary violations to already allocated operations. These problems are well known and the implications are discussed in [9] (for a general discussion see [6]).

In the model proposed here, we overcome the problems of conflict resolution through an approach called behaviour-based scheduling (BBS), where the individual entities assume an adaptive behaviour. The adaptation is implemented as a learning mechanism that continuously adjusts the time horizon of the jobs for which an entity may commit itself to other entities. By doing this, each entity manages a conflict free schedule in the short term while letting conflicts remain unresolved in the longer term. The temporal region in which there are no conflicts is called the commitment horizon. However, as shown in Figure 2, conflicts remain unresolved beyond this horizon. In fact, each individual entity may maintain a different commitment horizon, the length of which is purely determined by the extent of changes the entity experiences to its local schedule. These changes are brought about by the conflicts that are propagated by the entities and thus mutually influence the level of unresolved conflicts of the individual entity, and thus the conflicts at the system level. This way, when there is a high level of perturbations, the commitment horizon of the entities may shrink, whereas a stable situation may resolve more conflicts and the individual and overall schedules may become more predictable.

Apart from this adaptive mechanism, the model incorporates features for dynamically revising ADWs for operations as mentioned before, algorithms to allocate the operations, and provision for communications. For a detailed description see [9, 11]. The model used here extends some of these features, notably forward and backward communications of disturbances.

5. DESIGN OF EXPERIMENT

Heuristic approaches do not guarantee to produce optimal solutions. Naturally, therefore, BBS (which is a heuristic

based approach to scheduling) cannot in general be expected to provide the same or even similar levels of performance of mathematical optimisation techniques such as MIP. Nevertheless, of interest here is to ascertain, with some degree of confidence, the percentage range of MIP performance which BBS is able to achieve.

In this experiment both makespan and tardiness are chosen as the target performance measures for comparison between BBS and MIP. The model parameters of interest are the impact on performance due to variations in processing time of jobs at each station and the order due date factor. The former may affect the extent of imbalance in processing among the stations and hence the efficiency of coordination. Due date factor determines the slack available for the order and hence the tardiness performance and the scheduling efficiency.

Four categories of test data comprising of 120 problem sets each with 10 orders are created. Processing times are generated using normal and uniform distributions. Both fixed and variable due dates are generated. The fixed date is set arbitrarily at 250 days and for the variable date the length of the critical path is multiplied by a randomly generated factor between 1 and 5.

As BBS does not lend equal weight to minimising both makespan and tardiness, the objective function of MIP is set up to minimise tardiness. The performance between the approaches are measured as normalised paired differences with respect to MIP values.

6. ANALYSIS OF RESULTS

Computer run-time performance was not specifically monitored, suffice it to state however that, BBS final schedules for the 1200 orders were completed in a matter of hours (elapsed time) while MIP by comparison took approximately 13 days! The MIP computational performance was subsequently improved substantially [9].

In essence the tardiness performance results do not reveal any major surprises given that MIP is a mathematical optimisation routine with its objective function set to minimise total tardiness. The data in Table 1 highlights relative performance improvement for BBS for both variable due date data (vis-á-vis fixed due date data) and for the data from the normal distribution (vis-á-vis uniform distribution data). In fact factorial design and interaction data shows that interactions exist between the repeated measure (i.e. scheduling system) and the between subjects factors (i.e. Distribution type and Due date).

Due Date Type	Measure	Distribution	
		Normal	Uniform
Fixed	Average (MIP-BBS)	-126.47	-271.73
	Average (MIP-BBS)/MIP	-1.74	-3.67
	Times BBS > MIP	30	30
	Times BBS = MIP[1]	0	0
	Times BBS < MIP	0	0
Variable	Average (MIP-BBS)	-68.23	-102.37
	Average (MIP-BBS)/MIP	-0.38	-2.21
	Times BBS > MIP	16	28
	Times BBS = MIP	14	2
	Times BBS < MIP	0	0

Table 1: Tardiness Performance Of MIP vs BBS

The effect of fixed versus variable due date appears multiplicative rather than additive. In the case of the variable due date/normal distribution data sufficient order slack in the problem sets allowed both BBS and MIP to reach solutions with total tardiness values of zero. Due to this, the confidence intervals for the variable due date category (shown in Table 2) could not be calculated.

Due Date Type	Measure	Distribution	
		Normal	Uniform
Fixed	Confidence Interval[2]	-2.26 to -1.22 (-1.74 ± 0.52)	-4.85 to -2.49 (-3.67 ± 1.18)
Variable	Confidence Interval	Not Calculable	Not Calculable

Table 2: Confidence Intervals

Finally, the data also revealed that in absolute terms, BBS has greater variability (as evidenced by the standard deviation values) than MIP. Coefficient of variation values however reveal a different picture as shown in Table 3. Relative variation of BBS is slightly less than that of MIP.

Due Date Type	Normal Distribution		Uniform Distribution	
	MIP	BBS	MIP	BBS
Fixed	0.4652	0.3635	0.4609	0.3965
Variable	NA	NA	NA	NA

Table 3: Coefficient of Variation For Tardiness Performance

NA = Not Applicable

[1] Includes values of zero e.g., for tardiness BBS=MIP=0

[2] 95% confidence interval yields z-values of ± 1.96

We found the tardiness performance of BBS to be reasonably close to MIP. The confidence intervals show that the range of the normalised difference in tardiness lies between 1.22 and 4.85 times that of MIP.

6.1.1 Makespan Performance

Due Date Type	Measure	Distribution	
		Normal	Uniform
Fixed	Average (MIP-BBS)	-36.73	-60.03
	Average (MIP-BBS)/MIP	-0.12	-0.19
	Times BBS > MIP	30	29
	Times BBS = MIP[1]	0	0
	Times BBS < MIP	0	1
Var.	Average (MIP-BBS)	50.00	122.17
	Average (MIP-BBS)/MIP	0.09	0.20
	Times BBS > MIP	14	4
	Times BBS = MIP	0	0
	Times BBS < MIP	16	26

Table 4: Makespan Performance of (MIP-BBS)

The data in Table 4 reveals a similar pattern to that highlighted for tardiness in the fixed versus variable and normal versus uniform distribution results (shown in Table 1). In particular, the variable due date categories of data reveal superior performance of BBS over MIP.

With respect to makespan performance, the data in Table 5 shows that BBS outperformed MIP for the variable due date category (both normal and uniform distributions) but not for the fixed due date category. An alternative non-parametric test, the Wilcoxon Signed Rank Test for paired samples, indicated BBS results produced better makespan than MIP solutions. However we note that the MIP objective function only aims at minimising tardiness.

Due Date Type	Measure	Distribution	
		Normal	Uniform
Fixed	Confidence Interval[2]	-0.15 to -0.09 (-0.12 ± 0.03)	-0.23 to -0.15 (-0.19 ± 0.04)
Variable	Confidence Interval	0.03 to 0.15 (i.e. 0.09 ± 0.06)	0.14 to 0.26 (i.e. 0.20 ± 0.06)

Table 5: Confidence Intervals Based on Normalised Values of Makespan

[1] Includes values of zero e.g. for tardiness BBS=MIP=0
[2] 95% confidence interval yields a z-value of 1.96

Coefficient of variation values are summarised in Table 6 below which shows relative variation of BBS and MIP to be approximately equal for the fixed due date category and with BBS variability improved over MIP for the variable due date category.

Due Date Type	Normal Distribution		Uniform Distribution	
	MIP	BBS	MIP	BBS
Fixed	0.0631	0.0654	0.0965	0.0901
Variable	0.2151	0.0594	0.2209	0.0669

Table 6: Coefficient of Variation Values For Makespan Performance

From this result, we can state that :

- BBS produced better makespan values than MIP when variable due dates are employed. The confidence intervals show that the range of the normalised difference in makespan for this category lies between 0.03 and 0.26 less than MIP.
- MIP produced better results in makespan than BBS when fixed due dates are employed. The confidence intervals show that the range of the normalised difference in makespan for this category lies between 0.09 and 0.23 less than BBS.

7. SUMMARY AND CONCLUSIONS

The study reported here provides very useful insights into the performance of BBS. An important outcome is the demonstration of the feasibility of distributed scheduling approaches for solving assembly problems. Along with this, the results obtained can serve as a useful benchmark for future developments of this approach.

The primary conclusion of this study is that the performance of BBS is comparable to that of MIP. The makespan performance of BBS, particularly for the variable due date category of data, is quite impressive. However, the same trends as for tardiness, with respect to degradation of performance, are evident for makespan values i.e. worst performance is achieved when the problem is tightly constrained and processing times follow the uniform distribution. BBS performance with respect to run time (i.e. speed of response) and scalability (i.e. size of scheduling problem) are without doubt an order of magnitude superior to MIP.

As for sensitivity of performance of BBS, the results clearly

show that relative degradation of performance can be expected when the problem is tightly constrained as is the case with fixed due dates vis-á-vis variable due dates. The distribution of processing times also appears to have a marked influence with performance under the uniform distribution being significantly worse than for the normal distribution. Perhaps these types of issues can prove to be useful pointers for future revision of the BBS model.

Finally we may raise a point about distribution of scheduling capability. Normally, distribution may bring about simplification through decomposition of the problem space. Nevertheless, it can also be accompanied by complication in coordination of sub-problem solution that lead to conflicts. In this respect, BBS has demonstrated the possibility of overcoming this disadvantage.

REFERENCES

1. Agnetis, A., Pacifici, A., et al.,1997, "Scheduling of Flexible Flow Lines in an Automobile Assembly Plant", *European Journal of Operational Research*, 97, (2), 348-362.
2. Gou L., Hasegawa T., Luh P., Tamura S., and Oblak J.M., 1994, "Holonic Planning and Scheduling for a Robotic Assembly Testbed", *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, Try, NY, 10-12 October, IEEE Computer Society Press.
3. Kim, Y.D., Lim, H.G., Park, M.W.,1996, "Search Heuristics for a Flow-shop Scheduling Problem in a Printed Circuit Board Assembly Process", *European Journal of Operational Research*, **91**, 124-143.
4. Lassila O., Syrjanen M. and Torma S., 1990, "Coordinating Mutually Dependent Decisions in a Distributed Scheduler", *Conference on Advances in Production Management Systems 90*, Preprints, August 20-22, Dipoli, Espoo, Finland, pp. 1-8.
5. Lee, Chung-Yee, Cheng, T.C.E., Lin, B.M.T.,1993, "Minimising the Makespan in the 3-Machine Assembly-type Flow-shop Scheduling Problem", *Management Science*, **39**, (5), 616-625.
6. Lesser V.R., 1991, "A Retrospective View of FA/C Distributed Problem Solving", *IEEE Transactions on Systems, Man and Cybernetics*, v.**21**, n.6, Nov/Dec, pp. 1347-1362.
7. Muth, J.F. and Thompson, G.L. (eds.), 1963, "Industrial Scheduling", Prentice Hall Inc, New Jersey.
8. Ramaswamy S.E. and Joshi S.,1996, "Distributed Control of Automated Manufacturing Systems", Manufacturing Systems, v.**25**, n.4, *Proceedings of the CIRP Seminars*.
9. Tharumarajah A., 1995, PhD Thesis: "Scheduling Distributed Autonomous Manufacturing Systems", University of Melbourne, May.
10. Tharumarajah A. and Bemelman R., 1997, "Approaches and issues in scheduling a distributed shop-floor environment", *Computers in Industry*, **34**, 95-109.
11. Tharumarajah A. and Wells A.J.,1997, "A Behaviour-Based Approach to Scheduling in Distributed Manufacturing Systems", *Journal of Computer-Aided Engineering*, v.**4**, n.3.
12. Welgama, P., 1997, "Modelling and Assembly Line Scheduling Problem", *CMIS Technical Report*, CMIS-C **29/97**, CSIRO Mathematical and Information Sciences, North Ryde, Australia.