Short Papers

An Effective Method to Reduce Inventory in Job Shops

Peter B. Luh, Xiaohui Zhou, and Robert N. Tomastik

***Abstract*—Inventory plays a major role in deciding the overall manufacturing costs, and a good scheduling system should balance the on-time delivery of products versus low work-in-process (WIP) inventory. In this paper, the "CONstant work-in-process" (CONWIP) concept is applied to job shop scheduling to effectively control WIP inventory. A new mathematical formulation of CONWIP-based job shop scheduling with a separable structure is presented. By using a synergistic combination of Lagrangian relaxation, dynamic programming, and heuristic methods, good schedules are obtained in a reasonable amount of computation time. Results show that the new method can directly control the maximum WIP levels while maintaining good on-time delivery performance.**

***Index Terms*—Job shop, Lagrangian relaxation, manufacturing scheduling, work-in-process inventory.**

I. Introduction

Inventory plays an important role in scheduling because it has a major impact on overall manufacturing costs. Excessive inventory creates the needs for floor space, equipment, and manpower to transport, stock, and manage the inventory with no added value. Furthermore, defects are more difficult to detect, creating high rework and scraps. This study was motivated by a request from Sikorsky Aircraft, one of our industrial partners, to better control work-in-process (WIP) for job shops while maintaining good on-time delivery performance.

In most job shops, the "push-based" material requirement planning (MRP) is used as a production planning and scheduling tool. The WIP of these shops, however, is usually high mainly because of long lead times of MRP to handle shop floor uncertainties and the ignorance of machine capacities during the planning process. The "pull-based" just-in-time (JIT) production control manages WIP but is only applicable for repetitive manufacturing. Most other scheduling systems do not directly control inventory. Recently, the "CONstant work-in-process" (CONWIP) concept receives significant attention because of its simplicity in implementation, effectiveness in inventory control, and a few other advantages. Originally introduced for serial production lines, the strict CONWIP requires the number of parts simultaneously in the system to be equal to a certain constant—the maximum level of WIP allowed. It is argued that the CONWIP system is more effective than MRP and is more widely applicable than JIT [7].

For CONWIP-based serial production lines, the first-come first-served (FCFS) rule is usually used as the dispatching rule at work centers, and the key issue is to sequence the releases of raw materials to adequately utilize the machines and increase system throughput. Several heuristic methods have been presented in the literature. A heuristic method of releasing the highest priority job first for a multistation network was presented in [8], and a static work balance sequencing rule was used in scheduling networks of queues in [3]. CONWIP production lines were modeled as a tandem queuing system with a constant WIP level (number of containers) in [4], and the releasing sequence was obtained based on mean throughput and flow time using mean value analysis (MVA). The quality of schedules obtained by heuristic methods, however, is difficult to quantify and may be far from satisfaction.

The CONWIP concept can in principle be applied to other manufacturing settings (e.g., job shop). For CONWIP-based job shop scheduling, not only the releasing sequence of raw materials is important, but also the sequences of remaining operations on various machines are critical. Consequently, the problem is more complicated, and very little result has been reported. Based on our previous work [2], [6], a new formulation for CONWIP-based job shop scheduling is presented in Section II . Unlike other existing models, the CONWIP constraints presented here are additive and maintain the separability of the overall formulation. These CONWIP constraints are treated as extended machine capacity constraints and are effectively handled by using Lagrangian relaxation as presented in Section III. Numerical testing presented in Section IV shows that the method can directly control the maximum WIP level while maintaining good on-time delivery performance.

II. Problem Formulation

Extending the formulation of [2], the CONWIP-based job shop scheduling is formulated as an integer optimization problem. Assume that there are H machine types, each containing one or several identical machines. These machine types are indexed by h, $h = 0, \cdots, H-1$. There are I parts, and part i, $0 \leq i \leq I-1$, has due date d_i and priority ω_i and consists of J_i nonpreemptive sequential operations. Operation j, $0 \leq j \leq J_{i-1}$, of part i is denoted as (i, j) and requires a machine of type h belonging to a given set of eligible machine types H_{ij} for a specified processing time p_{ijh}. The scheduling horizon consists of K time units, indexed by k $(k = 0, \cdots, K-1)$. The objective function and constraints of the formulation are presented below.

A. Objective Function

The goals of on-time delivery and low WIP inventory are modeled as weighted penalties on tardiness and on releasing raw materials too early (the earliness penalties) in the objective function, i.e.,

$$J \equiv \sum_{i=0}^{I-1} \omega_i T_i^2 + \sum_{i=0}^{I-1} \beta_i E_i^2. \tag{2.1}$$

In the above, the tardiness T_i is the amount of overdue time, i.e., max $(c_i - d_i, 0)$ with c_i the part completion time (the completion time of the part's last operation) and d_i its due date. Earliness E_i is defined as the amount that part beginning time (the beginning time of the part's first operation) leads the desired release time following [2]. The purpose of tardiness penalties is to maintain good on-time delivery performance, while the purpose of earliness penalties is to prevent releasing raw materials too early, indirectly controlling WIP inventory. The weights are selected based on the following guidelines: the tardiness weight ω_i of part i reflects the importance or priority of the

Manuscript received February 26, 1999; revised January 27, 2000. Paper approved for publication by Associate Editor Y. Narahari and Editor N. Viswanadham upon evaluation of the reviewers' comments. This research was supported in part by the National Science Foundation under Grant DMI-9500037 and Grant DMI-9813176, and in part by the United Technologies Research Center. This paper was presented at the IEEE International Conference on Robotics and Automation, Detroit, MI, May 1999.

P. B. Luh and X. Zhou are with the Department of Electrical and Systems Engineering, University of Connecticut, Storrs, CT 06269-2157 USA.

R. N. Tomastik is with the United Technologies Research Center, East Hartford, CT 06108 USA.

Publisher Item Identifier S 1042-296X(00)06954-8.

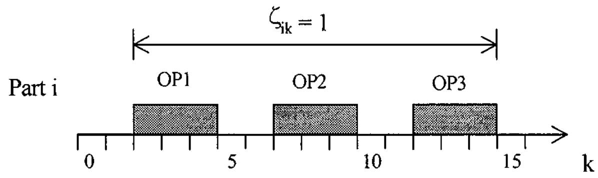

Fig. 1. An illustration of an inventory variable.

part (generally 1–10), and the earliness weights β_i is inversely proportional to the corresponding tardiness weight. This is because a part with higher priority should be allowed to start earlier to meet its due date, and the earliness penalty should be the secondary as compared to the tardiness penalty.

B. Machine Capacity Constraints

The machine capacity constraints state that the total number of operations assigned to machine type h must be less than or equal to the number of machines available at any time, i.e.,

$$\sum_{i=0}^{I-1} \sum_{j=0}^{J_i-1} \delta_{ijkh} \leq M_{kj}, \qquad k = 0, \cdots, K-1;\ h \in H \tag{2.2}$$

where δ_{ijkh} is a 0–1 integer variable equal to 1 if operation (i, j) is processed on machine type h at time k, and 0 otherwise, i.e.,

$$\delta_{ijkh} = \begin{cases} 1, & b_{ij} \leq k \leq c_{ij} \\ 0, & \text{otherwise} \end{cases}$$

where b_{ij} and c_{ij} are the beginning and completion times of operation (i, j), respectively.

C. CONWIP Constraints

For serial production lines, the CONWIP constraints are usually modeled by having a fixed number of "containers" or "cards" in the system, and when both a new material and a vacant container or card are available, the new material will be released into the system to make the overall WIP constant. For job shops, a part is counted as WIP after it is released to the shop up to its completion, i.e., from the beginning of its first operation to the completion of its last operation. A set of 0–1 integer "inventory variables" $\{\zeta_{ik}\}$ is therefore introduced. It equals 1 when part i is in the shop at time k, and 0 otherwise, i.e.,

$$\zeta_{ik} \begin{cases} 1, & b_{i,0} \leq k \leq c_{i,J_i-1} \\ 0, & \text{otherwise}. \end{cases}$$

An inventory variable of part i with release time equal to 2 and completion time 14 is illustrated in Fig. 1.

The strict constant WIP control policy for serial production lines may not be good for job shops. This is because due dates of parts may vary widely, and releasing materials too early to satisfy the strict CONWIP constraints may result in higher WIP and higher cost because of the existence of earliness penalties. Therefore, the CONWIP-based job shop requires the number of parts simultaneously in the shop to be less than or equal to W, the maximum WIP inventory allowed at any time, i.e.,

$$\sum_{i=0}^{I-1} \zeta_{ik} \leq W, \qquad k = 0, \cdots, K-1. \tag{2.3}$$

D. Operation Precedence Constraints

An operation cannot be started until its preceding operation has been finished, i.e.,

$$c_{ij} + 1 \leq b_{i,j+1} \qquad \forall (i, j) \tag{2.4}$$

where c_{ij} and $b_{i,j+1}$ are the completion time of operation (i, j) and the beginning time of operation $(i, j+1)$, respectively. The term "1" is required in (2.4) since operation (i, j) is assumed to be completed at the end of time unit c_{ij}, and operation $(i, j+1)$ is assumed to be started at the beginning of time unit b_{ij}. For the same reason, the term "1" also appears in the following processing time requirements.

E. Processing time Requirements

Each operation must be assigned the required amount of processing time on a selected machine type h belonging to the given set of eligible machine types H_{ij}, i.e.,

$$c_{ij} = b_{ij} + p_{ijh} - 1 \qquad \forall (i, j); h \in H_{ij} \tag{2.5}$$

The overall problem is to minimize the objective function (2.1) subject to constraints (2.2)–(2.5) by selecting the machine types $\{h_{ij}\}$ and beginning times $\{b_{ij}\}$ for all operations. Once $\{b_{ij}\}$ and $\{h_{ij}\}$ are selected, $\{c_{ij}\}$, $\{T_i\}$, $\{E_i\}$, $\{\delta_{ijkh}\}$, and $\{\zeta_{ik}\}$ can be easily derived. Since all the constraints are linear and the objective function is additive, the problem is "separable," which is essential for Lagrangian relaxation to be effective.

III. Solution Methodology

Machine capacity constraints and CONWIP constraints are first relaxed by using Lagrangian multipliers. The "relaxed problem" can then be decomposed into smaller and easier part subproblems. These subproblems are solved by using backward dynamic programming (BDP) with stages corresponding to operations, precedence constraints embedded in state transitions, and the "inventory" component of the part costs (WIP cost) efficiently calculated by treating CONWIP as an "extended" machine type. Finally, a heuristic method is developed to generate feasible schedules satisfying all the constraints based on subproblem solutions.

A. Lagrangian Relaxation

By using Lagrangian multipliers π_{kh} to relax machine capacity constraints and ρ_k the CONWIP constraints, the following relaxed problem is obtained:

$$\begin{aligned} \min L,\ \text{with } L \equiv & \sum_i \left(\omega_i T_i^2 + \beta_i E_i^2\right) \\ & + \sum_i \sum_{jkh} \pi_{kh} \delta_{ijkh} - \sum_{kh} \pi_{kh} M_{kh} \\ & + \sum_i \sum_k \rho_k \zeta_{ik} - \sum_k \rho_k W \end{aligned} \tag{3.1}$$

subject to (2.4) and (2.5). By regrouping relevant terms, the relaxed problem can be decomposed into the following part-level subproblems:

$$\begin{aligned} \min_{\{b_{ij}, h_{ij}\}} L_i,\ \text{with } L_i \equiv & \ \omega_i T_i^2 + \beta_i E_i^2 \\ & + \sum_{j=0}^{J_i-1} \sum_{k=b_{ij}}^{c_{ij}} \pi_{kh} + \sum_{k=b_{i0}}^{c_i J_i - 1} \rho_k \end{aligned} \tag{3.2}$$

subject to (2.4) and (2.5). The above cost L_i includes tradiness penalty, earliness penalty, the cost for using machines, and the cost for contributing to WIP.

B. Dynamic Programming

In solving part subproblems, the backward DP is used, with stages corresponding to operations and precedence constraints embedded in state transitions following [1] and [5]. Since inventory variables $\{\zeta_{ik}\}$

involve queue times as well as operation processing times, the calculation of the WIP cost within BDP is not easy.

From another point of view, CONWIP can be treated as an extended machine type with capacity equal to W. The major difference between this extended machine type and a regular machine type is that a "WIP" operation does not have a "fixed processing time." It starts as a part is released to the shop and completes when its last operation is finished as explained earlier. Since undetermined queue times are involved, the completion time cannot be uniquely calculated for a given beginning time in advance. Neverthless, the following identity holds:

$$\sum_{k=b_{i0}}^{c_{i,J_i=1}} \rho_k = \sum_{k=0}^{c_{i+J_i-1}} \rho_k - \sum_{k=0}^{b_{i,0}-1} \rho_k. \tag{3.3}$$

Consequently, the WIP cost can be allocated to the first and the last operation, and be easily calculated.

The BDP algorithm starts from the last stage with the following terminal cost:

$$V_{i,J_i-1}(b_{i,J_i-1}, h_{i,J_i-1}) = \omega_i T_i^2 + \sum_{k=b_{i,J_i-1}}^{c_{i,J_i-1}} \pi_{kh_{i,J_i-1}} + \sum_{k=0}^{c_{i,J_i-1}} \rho_k. \tag{3.4}$$

The cumulative cost when moving backward is calculated recursively by

$$\begin{aligned} V_{i,j}(b_{ij}, h_{ij}) &= \min_{\{b_{i,j+1}, h_{i,j+1}\}} \left(\left(\beta_i E_i^2 - \sum_{k=0}^{b_{i,0}-1} \rho_k \right) \Delta_i \right. \\ &\quad \left. + \sum_{k=b_{ij}}^{c_{ij}} \pi_{kh} + V_{i,j+1}(b_{i,j+1}, h_{i,j+1}) \right) \\ &= \left(\beta_i E_i^2 - \sum_{k=0}^{b_{i,0}-1} \rho_k \right) \Delta_i + \sum_{k=b_{ij}}^{c_{ij}} \pi_{kh} \\ &\quad + \min_{\{b_{i,j+1}, h_{i,j+1}\}} (V_{i,j+1}(b_{i,j+1}, h_{i,j+1})), \\ &\qquad 0 \le j \le J_i - 2 \end{aligned} \tag{3.5}$$

where Δ_i is an integer variable equal to one if operation (i, j) is the first operation of part i, and zero otherwise. The optimal L_i^* is obtained as the minimal cumulative cost at the first stage, and the optimal beginning times and the corresponding selected machine types can be obtained by tracing forward the stages.

C. Dual Problem

Given the optimal subproblem costs $\{L_i^*\}$, the high level dual problem is obtained as

$$\min_{\{\pi_{kh}, \rho_k\}} D,$$

$$\text{with } D = \sum_i L_i^* - \sum_{kh} \pi_{kh} M_{kh} - \sum_k \rho_k W \tag{3.6}$$

$$\text{subject to } \pi_{kh} \ge 0, \qquad k = 0, \cdots, K-1; h \in H \tag{3.7}$$

$$\text{and } \rho_k \ge 0, \qquad k = 0, \cdots, K-1. \tag{3.8}$$

In this study, the subgradient method is used to iteratively update Lagrangian multipliers in solving the dual problem.

D. Obtaining Feasible Solution

Since the machine capacity constraints and the CONWIP constraints are relaxed, the solutions for part subproblems, when put together, may not be feasible. With the updating of multipliers at the high level, the part subproblem solutions will be more and more feasible. If the dual problem solution can converge to its optimal, the corresponding optimal part subproblem solutions, when put together, is an optimal primal solution when there is no duality gap. When there is an inherent duality gap, the optimal part subproblem solutions may be infeasible, but they will provide valuable information for the construction of the feasible primal solution in heuristics.

A modified version of the list scheduling heuristics of [6] is developed to generate a feasible schedule based on subproblem solutions. In the heuristics, operations are first arranged in a list in the ascending order of their subproblem beginning times. They are then assigned to machines according to this list as machines become available. When operations have same beginning times, they are sorted in the list according to the incremental cost as defined in [6], and the operation with higher incremental cost will be scheduled first. In addition, the WIP level is computed for each time k, and only when the WIP level is smaller than W can a new part be released. Otherwise, new parts are delayed to the next time slot. The process is repeated until all operations are assigned, and the cost for this feasible schedule can be computed. This heuristics can be run many times as the multipliers are updated, and the schedule with the lowest cost is chosen as the final schedule.

IV. Numerical Results

The method has been implemented in C++ on a Pentium II 400-MHz personal computer. Three examples are presented below to demonstrate the method and to present insights obtained. In the testing, all the multipliers are initialized to zero, and the subgradient algorithm terminated after a fixed number of iterations (Examples 1 and 2), or after a fixed amount of CPU time for the larger Example 3. In addition, the "time step reduction technique" is used to handle the long time horizon for Examples 2 and 3 following [6].

A. Example 1

Eleven data sets were randomly generated in this example to assess the impact of CONWIP levels. There are 10 to 14 machines and 100 parts for each data set. For each part, there are two to five operations. The tardiness weight equals 1 and the earliness weight equals 0.5. The machine types, operation processing times, and part due dates were randomly generated with discrete uniform distributions within appropriate intervals.

The job shop scheduling results without CONWIP constraints were obtained first, and the maximum WIP was recorded for each case. Then, the CONWIP-based job shop scheduling results were obtained for different CONWIP levels (W was set as a fraction C times the maximum WIP and rounded up to the next integer). The detailed job shop scheduling results without CONWIP constraints are presented in Table I.

Among the 11 data sets, the first 10 have high machine utilization (around 80%). They reflect our past experience that, in reality, many parts can be overdue and, therefore, should be processed as soon as possible. The duality gaps of these ten cases are illustrated in Fig. 2. Higher duality gap is expected for CONWIP-based job shop because the additional CONWIP constraints are relaxed. It can also be seen from the figure that when CONWIP constraints are loose $(C = 0.8)$, the schedule quality measured in terms of duality gap is good (the average duality gap is 24.06%) as it is not too difficult to satisfy the CONWIP constraints. When CONWIP constraints are very tight $(C = 0.2)$, the schedule quality is also good (the average duality gap is 24.06%). This is because the multipliers relaxing CONWIP constraints dominate the solution process, whereas the multipliers relaxing machine capacity constraints are mostly inactive. If these two types of constraints are both active $(C = 0.6$ and $C = 0.4)$, relatively larger gaps (39.58%

TABLE I
NUMERICAL RESULTS FOR EXAMPLE 1

Case	Dual Cost	Feasible Cost	Duality Gap %	Max. WIP	Aver. WIP	Aver. Mach. Util. %
1	121146	154729	27.72	23	13.66	74.78
2	58849	75579	28.43	18	13.32	85.02
3	57249	63611	11.11	18	13.64	87.38
4	212394	238356	12.22	25	16.11	82.11
5	53407	61589	15.32	25	15.78	77.91
6	250571	308471	23.11	24	15.39	85.56
7	41991	46950	11.81	25	16.51	80.03
8	80574	89363	10.91	21	13.0	82.45
9	150941	167022	10.65	29	15.81	79.01
10	27489	31587	14.91	21	12.05	76.64
11	218	236.5	8.4	9	4.71	44.87

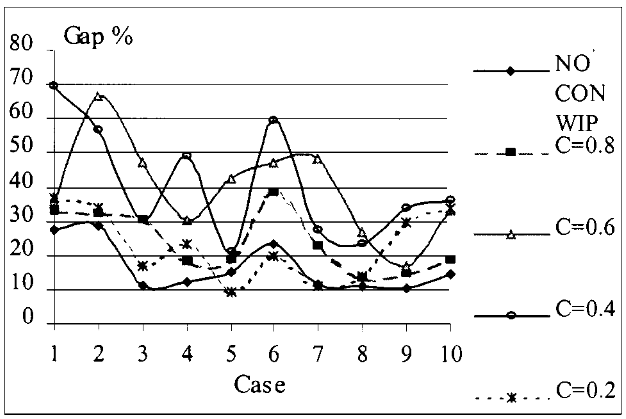

Fig. 2. Duality gaps for cases 1–10.

TABLE II
COSTS FOR CASE 1 AND CASE 11 IN EXAMPLE 1

		NO CON WIP	C = 0.8	C = 0.6	C = 0.4	C = 0.2
Case 1	Dual cost	121146	119247	118312	112101	389556
	Feasible cost	154729	158010	160577	189902	532484
Case 11	Dual cost	218	200.03	288.84	4787	513710
	Feasible cost	236.5	257	850	18340.5	619309

and 40.71%, respectively) are encountered. The feasible costs do increase as C decreases, as illustrated by the third row of Table II for Data Set 1.

To explore the performance of this algorithm under low machine utilization situation, the 11th data set is generated based on Data Set 1 by reducing all the operation processing times to half and then rounded up to the next integer value. Material arrival times are generated based on a Poisson distribution, and part due dates are set to be the material arrival times plus the due dates of Data Set 1. The machine utilization is decreased to an average of 44.9%. The scheduling costs for different CONWIP levels are presented in the fourth and fifth row of Table II. Compared with high utilization results, the similar trend is observed. Since most parts can satisfy their due dates and Lagrange Relaxation is a due date driven method, the costs are much less than those of Case 1, resulting in higher relative gaps.

TABLE III
NUMERICAL RESULTS FOR EXAMPLE 2

Case	CONWIP in Opti.	CONWIP in Heuristics	Avg. WIP	Dual Cost	Feasible Cost	Duality Gap
0	N	N	51.72	239513	274788	14.87
1	Y	Y(W=60)	40.53	247448	292919	18.38
2	N	Y (W=60)	40.70	240715	291630	N/A
3	Y	Y (W=50)	33.58	281426	343870	22.24
4	N	Y (W=50)	33.36	239413	361772	N/A
5	Y	Y(W=40)	31.74	358654	435296	21.37
6	N	Y(W=40)	31.43	236515	480648	N/A

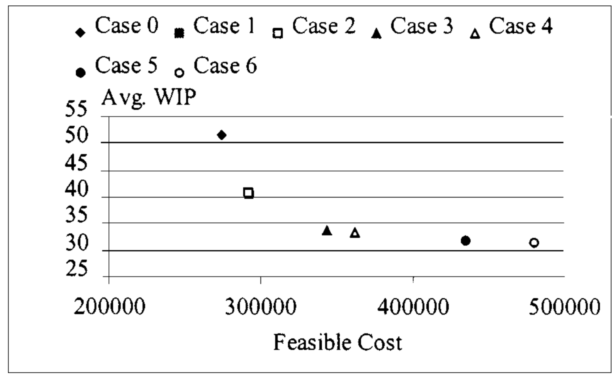

Fig. 3. Plot of average WIP versus feasible costs.

B. Example 2

This example is to illustrate the effects of having CONWIP constraints in optimization. There are 5 different machines and 100 parts. Each part has 2–12 operations, and the weight settings are the same as in Example 1. Two methods are tested: one with the CONWIP constraints applied in both optimization and heuristics, and the other with the CONWIP constraints only included in heuristics. Scheduling results for different CONWIP levels are presented in Table III.

Although the average WIP levels shown in Table III are below the maximum allowed levels, the maximum WIP levels were reached many times for all the cases. From Fig. 3, it can be seen that having the CONWIP constraints in optimization significantly reduced feasible costs as compared to the cases that CONWIP constraints were only included in heuristics, and the effect is drastic as W decreases. It is also observed that the feasible cost J becomes larger as W decreases because more parts are delayed, and the rate of increase is steep as W reduces below a certain threshold. This threshold, however, may not be apparent by simply looking at the problem, and the tradeoff between on-time delivery and low WIP inventory may not be easy without such results.

C. Example 3

This data set was created by randomly picking 180 parts from a realistic database. There are 14 different machines and each part with 2–15 operations for a total of 1504 operations.initial WIP (parts already in the shop before scheduling) is 30, most parts are due very early, and 10 parts are due very late (around 70 days after the scheduling beginning time). Two cases were tested to examine the effectiveness of CONWIP constraints in controlling WIP. The algorithm stopped after 10 min of CPU time, and the results are presented in Table IV.

In Case 1, there are no CONWIP constraints, and the WIP over the scheduling horizon is depicted in Fig. 4. The WIP levels are very high

TABLE IV
NUMERICAL RESULTS FOR EXAMPLE 3

Case	CONWIP Constr.	Max WIP	Average WIP	Dual Cost x1000000	Feasible Cost x1000000	Duality Gap
Case 1	N	90	28.58	381.8	431.5	13.02%
Case 2	Y(W=40)	40	22.57	391.67	452.28	15.47%

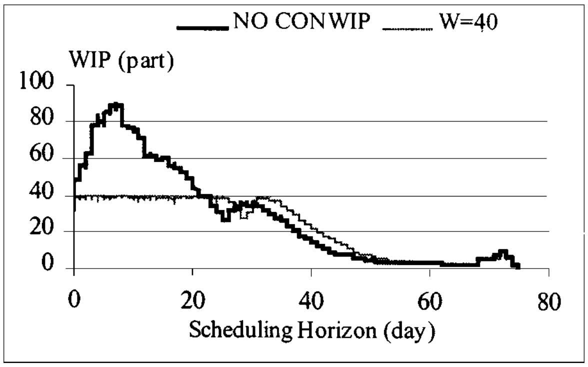

Fig. 4. WIP distribution for Example 3.

for the first 20 days because parts that have very early due dates are released to fully utilize the machines, regardless of the WIP level. The parts with very late due dates are not released in view of the existence of earliness penalties. The solution quality, as measured in terms of duality gap, however, is good for this case.

In Case 2, CONWIP constraints with $W = 40$ were added, and the WIP over the scheduling horizon is overlaid in Fig. 4. Compared to Case 1, the WIP levels are much lower. At the same time, the feasible cost increases only slightly because the CONWIP constraints were included in optimization, and the resulting schedule maintains good on-time delivery performance. The maximum WIP decreased drastically for the first 20 days and the delay on part completion is very small.

V. CONCLUSIONS

A new separable formulation for CONWIP-based job shop scheduling is presented. A Lagrangian relaxation-based algorithm is developed to solve the problem. Testing results demonstrate that it can generate schedules with controllable WIP levels. At the same time, good on-time delivery performance is obtained.

The formulation can also be easily extended to cover other situations. For example, different part types may vary in size and value, and it may be more reasonable to have different weights for different part types in calculating WIP levels. This constant weighted WIP concept can be easily formulated by having weights for parts in (2.3). Another example is that a job shop may want to limit WIP levels down to individual part types or families of part types as opposed to having one WIP level for the entire factory. This can be similarly modeled by having the CONWIP constraints for selected part types or families of part types.

ACKNOWLEDGMENT

The authors would like to thank Y. Zhang and X. Zhao at the University of Connecticut for their assistance and invaluable suggestions.

REFERENCES

[1] H. Chen, C. Chu, and J. M. Proth, "A more efficient Lagrangian relaxation approach to job shop scheduling problems," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1995, pp. 496–501.

[2] C. Czerwinski and P. B. Luh, "Scheduling products with bills of materials using an improved Lagrangian relaxation technique," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 99–111, 1994.

[3] I. Duenyas, "A simple release policy for networks of queues with controllable inputs," *Oper. Res.*, vol. 42, no. 6, pp. 1162–1171, 1994.

[4] Y. T. Herer and M. Masin, "Mathematical programming formulation of CONWIP-based production lines and relationship to MRP," *Int. J. Prod. Res.*, vol. 35, no. 4, pp. 1067–1076, 1997.

[5] J. Wang, P. B. Luh, X. Zhao, and J. Wang, "An optimization-based algorithm for job shop scheduling," *Sadhana*, vol. 22, no. 2, pp. 241–256, 1997.

[6] P. B. Luh, L. Gou, Y. Zhang, T. Nagahora, M. Tsuji, K. Yoneda, T. Hasegawa, Y. Kyoya, and T. Kano, "Job shop scheduling with group-dependent setups, finite buffers, and long time horizon," *Ann. Oper. Res.*, vol. 76, pp. 233–259, 1998.

[7] M. L. Spearman, D. L. Woodruff, and W. J. Hopp, "CONWIP: A pull alternate to Kanban," *Int. J. Prod. Res.*, vol. 28, no. 5, pp. 679–894, 1990.

[8] L. M. Wein, "Scheduling network of queues: Heavy traffic analysis of a multi-station network with controllable input," *Oper. Res.*, vol. 40, no. 4, pp. S312–S334, 1992.

Deadlock Avoidance in Flexible Manufacturing Systems Using Finite Automata

Ali Yalcin and Thomas O. Boucher

***Abstract*—A distinguishing feature of a flexible manufacturing system (FMS) is the ability to perform multiple tasks in one machine or workstation (alternative machining) and the ability to process parts according to more than one sequence of operations (alternative sequencing). In this paper, we address the issue of deadlock avoidance in systems having these characteristics. A deadlock-free and maximally permissive control policy that incorporates this flexibility is developed based on finite automata models of part process plans and the FMS. The resulting supervisory controller is used for dynamic evaluation of deadlock avoidance based on the remaining processing requirements of the parts.**

***Index Terms*—Deadlock avoidance, finite automata, flexible manufacturing systems, supervisory control.**

I. INTRODUCTION

A typical flexible manufacturing system (FMS) is composed of single machines or workstations that can perform various operations and a material handling system that interconnects these machines. Raw parts enter the system at discrete points in time. These parts are

Manuscript received January 6, 1999; revised October 14, 1999. This paper was approved for publication by Associate Editor Y. Narahari and Editor P. Luh upon evaluation of the reviewers' comments.

A. Yalcin was with with the Department of Industrial Engineering, Rutgers University, Piscataway, NJ 08854 USA. He is now with the Department of Industrial and Management Systems Engineering, University of South Florida, Tampa, FL 33620 USA.

T. O. Boucher is with with the Department of Industrial Engineering, Rutgers University, Piscataway, NJ 08854 USA.

Publisher Item Identifier S 1042-296X(00)06949-4.