Proceedings of the 29th Conference
on Decision and Control
Honolulu, Hawaii • December 1990

FLOW SHOP SCHEDULING BY A LAGRANGIAN RELAXATION AND NETWORK FLOW APPROACH*

Shi-Chung Chang, Da-Yin Liao, Fu-Shiung Hsieh
Dept. of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R. O. C.

Chi-Tsai Yang
Information and Administration Dept.
Fu Sheng Industrial Co.
San-Chung, Taiwan, R. O. C.

ABSTRACT

This paper develops, by using advanced optimization methods, a production scheduling algorithm for discrete–part, make–to–order type of flexible flow shops. The goal of scheduling is to meet due dates and the problem is formulated as a large–scale integer programming problem. Our scheduling algorithm includes four parts : decomposition by Lagrangian relaxation into subproblems, the minimum cost linear network flow algorithm for solving subproblems, a subgradient algorithm for solving the dual problem, and a heuristic to obtain a near–optimal and feasible schedule. The algorithm is then compared with a heuristic rule that is considered effective by some local manufacturing firms. Preliminary results show that our scheduling algorithm not only is better in optimality but also provides more insights into scheduling complicated operations.

I. INTRODUCTION

Production scheduling is one of the most important issues in shop floor control of a manufacturing firm. Although there have been numerous researches on this topic in the literature [4], there are needs for new concepts and advanced techniques to schedule manufacturing operations as many new technologies have been developed and installed in manufacturing firms today [7]. In this paper, we consider a make–to–order flow shop which manufactures discrete parts of different types. Each type of parts have its own due date and desired quantity. Machine groups of limited processing capacities with infinite intermediate buffer sizes are used to manufacture these parts. Each machine group consists of homogeneous machines. There is one production process associated with each part type which consists of a sequence of operations requiring different machining facilities. However, different part types may have processing operations that need a same machine group. Given a set of orders, we want to find a schedule that minimizes weighted production tardiness subject to constraints of (1) machine capacity and availability, (2) end product demand, and (3) precedence relationship of manufacturing processes.

We first formulate the above scheduling problem as an integer programming problem in Section II. In Section III, we describe our solution algorithm which is developed based on a Lagrangian relaxation and network flow approach. Some preliminary scheduling results of our algorithm are presented in Section IV. Its comparisons to the "Priority Factor" (PF) heuristic [3] are also given. Section V then briefly concludes our study.

II. PROBLEM FORMULATION

To convey our main idea and simplify the discussion, we assume that all types of parts require the same sequence of processing by machine groups in the flow shop, i.e., the M machine groups and the associated buffers can be organized as a line of production shown in Figure 2.1.

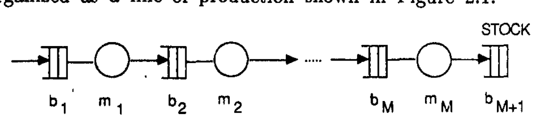

Figure 2.1

* This work was supported in part by the National Science Council of the Republic of China under Grants NSC–78–0422–E002–09 and NSC–79–0422–E002–05.

Let us now define some notations for describing the above flow shop.

Notations

I :total number of part types;
i :part type index, $i=1, \cdots, I$;
D_i :demand of type i parts;
d_i :due date of type i parts;
M :total number of machine groups;
m :machine group index, $m = 1, \cdots, M$;
C_m :capacity of machine group m;
P_{im} :processing time of unit type i part on machine group m;
t :time index;
b :buffer index, where b=m for the buffer before machine group m, and b=M+1 for the stock of finished parts;
X_{ibt} :number of type i parts in buffer b at the beginning of time period t;
u_{imt} :number of type i parts loaded onto machine group m for processing at the beginning of period t;
Z_{it} :number of type i parts arriving at the stock at time period t, where $Z_{it} = u_{iM(t-P_{iM})}$.

Assume that it takes the same amount of processing time as long as the number of type i parts being processed is within the capacity of machine group m. Also assume that all the production demands $\{D_i\}$ are released at the beginning of scheduling. In such a shop, a batch of type i parts loaded onto machine (m–1) for processing at period $t-P_{i(m-1)}$ go into buffer m after $P_{i(m-1)}$ periods of processing at machine group (m–1). The stock buffer serves as a sink and accumulates finished parts. The flows of parts therefore must satisfy the following flow balance equations.

Flow Balance Equations

$$X_{i10} = D_i; \qquad (2.1.a)$$

$$X_{i1(t+1)} = X_{i1t} - u_{i1t}; \qquad (2.1.b)$$

$$X_{im(t+1)} = X_{imt} - u_{imt} + u_{i(m-1)(t-P_{i(m-1)})}, \quad \forall\ m\neq 1, M+1; \qquad (2.1.c)$$

$$X_{i(M+1)(t+1)} = X_{i(M+1)t} + u_{iM(t-P_{iM})}; \qquad (2.1.d)$$

for $i = 1, \cdots, I$.

Since a batch of u_{imt} parts loaded onto machine group m needs P_{im} periods to complete the processing, the quantity $\sum_{i=1}^{I} \sum_{\tau=t-P_{im}+1}^{t} u_{im\tau}$ is the total number of parts which is being processed on machine m during time period t. This quantity must not exceed the processing capacity of machine group m, i.e.,

Machine Capacity Constraints

$$\sum_{i=1}^{I} \sum_{\tau=t-P_{im}+1}^{t} u_{im\tau} \leq C_m, \qquad \forall\ t, m. \qquad (2.2)$$

Obviously, we should have the constraints that u_{imt} and X_{ibt} are nonnegative integers,

$$\forall\ i, m, b, t. \qquad (2.3)$$

Our objective of production scheduling is to meet due dates if possible. A penalty cost is incurred by an overdue production, which increases with production tardiness. We then formulate the production scheduling problem as follows:

$$\text{(P)} \qquad \min_{\mathbf{u}} \sum_{i=1}^{I} \sum_{t=1}^{T} \psi_{it} Z_{it}$$

subject to constraints (2.1–3),

$$\text{where } \psi_{it} = \begin{cases} 0, & t \le d_i, \\ A_i(t-d_i), & t > d_i, \end{cases}$$

and A_i is the overdue penalty cost coefficient.

III. SOLUTION ALGORITHM DEVELOPMENT

The production scheduling problem (P) formulated above is an integer programming problem of NP–hard computational complexity [6]. Here, we develop for it a computationally feasible, near–optimal solution algorithm. It consists of four parts :

(1) the dual problem formulation and decomposition into production scheduling subproblems according to part types by applying Lagrangian relaxation to the machine capacity constraint;
(2) application of a minimum cost network flow algorithm to solve each production scheduling subproblem;
(3) application of a simple nondifferentiable optimization scheme to the dual problem, and
(4) development of a heuristic algorithm that finds a near–optimal, feasible solution based on the solution of the relaxed problem and the network structure.

III.1 Lagrangian Relaxation and Decomposition

In problem (P), we observe that coupling among flows of different part types is through the machine capacity constraints. Applying Lagrange Relaxation to constraint (2.2), we form the Lagrangian function as

$$\sum_{i=1}^{I} \sum_{t=1}^{T} \psi_{it} Z_{it} + \sum_{t=1}^{T} \sum_{m=1}^{M} \lambda_{mt} \Big(\sum_{i=1}^{I} \sum_{\tau=t-P_{im}+1}^{t} u_{im\tau} - C_m \Big) \tag{3.1.a}$$

$$= \sum_{i=1}^{I} \sum_{t=1}^{T} \Big(\psi_{it} Z_{it} + \sum_{m=1}^{M} \lambda_{mt} \sum_{\tau=t-P_{im}+1}^{t} u_{im\tau} \Big) - \sum_{t=1}^{T} \sum_{m=1}^{M} \lambda_{mt} C_m \tag{3.1.b}$$

where λ_{mt} is the associated Lagrange multiplier.

Since the function above is additively separable in $\mathbf{u}_i$ for a given set of Lagrange multipliers $\boldsymbol{\lambda}$, we define

$$L_i(\mathbf{u}_i, \boldsymbol{\lambda}) \equiv \sum_{t=1}^{T} \Big(\psi_{it} Z_{it} + \sum_{m=1}^{M} \lambda_{mt} \sum_{\tau=t-P_{im}+1}^{t} u_{im\tau} \Big) \tag{3.2}$$

and the dual function

$$\Phi(\boldsymbol{\lambda}) \equiv \sum_{i=1}^{I} \min_{\mathbf{u}_i} L_i(\mathbf{u}_i, \boldsymbol{\lambda}) - \sum_{t=1}^{T} \sum_{m=1}^{M} \lambda_{mt} C_m \tag{3.3}$$

subject to (2.1) and (2.3).

The dual problem is then

$$\text{(D)} \qquad \max_{\boldsymbol{\lambda} \ge 0} \Phi(\boldsymbol{\lambda}),$$

and the scheduling subproblem for type i parts is

$$\text{(P–i)} \qquad \min_{\mathbf{u}_i} \Big[L_i(\mathbf{u}_i, \boldsymbol{\lambda}) \equiv \sum_{t=1}^{T} \Big(\psi_{it} Z_{it} + \sum_{m=1}^{M} \lambda_{mt} \sum_{\tau=t-P_{im}+1}^{t} u_{im\tau} \Big) \Big]$$

subject to (2.1) and (2.3) for type i only, which is independent of the scheduling for other types.

III.2 A Network Model

Each scheduling subproblem (P–i) can be formulated as a minimum cost network flow problem due to the network structure of flow balance equations (2.1). The network, which is a graph consisting of nodes and arcs, is constructed as follows. Let a node n_{mt} represent buffer m at time t, $m=1, \cdots, M, M+1$ and $t=1,\cdots, T$. Since it takes P_{im} periods of processing time for a part to go from buffer m to buffer m+1, we form an arc $(n_{mt}, n_{(m+1)(t+P_{im})})$ to represent a flow path between the two buffers. The remaining parts in buffer m after loading u_{imt} for processing at t is carried by buffer m over to time period t+1, i.e., they flow through arc $(n_{mt}, n_{m(t+1)})$. In addition, we let nodes S and T be the source and sink nodes of the network respectively. The source node **S** is connected by an arc to node n_{11} with a constant flow equal to the demand D_i. The terminal node **T** serves as a sink of flows. The flow on arc $(n_{mt}, n_{(m+1)(t+P_{im})})$ is u_{imt} and Z_{it} on arc $(n_{(M+1)t}, T)$. It can be easily checked out that the flow conservation at each node represents one of the flow balance equations. We can find the associated arc cost for arc $(n_{mt}, n_{(m+1)(t+P_{im})})$ by appropriately rearranging terms in subproblem (P–i) as $\sum_{\tau=t}^{t+P_{im}-1} \lambda_{m\tau}$, $\forall$ t and m. The cost of arc $(n_{(M+1)t}, T)$ is ψ_{it}. There is no cost for flows on arc $(n_{mt}, n_{m(t+1)})$. From the above discussion, subproblem (P–i) is now clearly formulated as a minimum cost linear network flow (MCLNF) problem, which has an integer optimal solution. We adopt the RELAX code of Bertsekas and Tseng [1] to solve it.

III.3 Solving the Dual Problem

Let $\{\mathbf{u}_i^*\}$ be the optimal solution to subproblems for the set of Lagrange multipliers $\boldsymbol{\lambda}^k$ at kth iteration. Since $\Phi(\boldsymbol{\lambda})$ is nondifferentiable with respect to $\boldsymbol{\lambda}$, we adopt a subgradient scheme of [2] to solve (D). The scheme is summarized as follows :

The gradient of $\Phi(\boldsymbol{\lambda})$ with respect to λ_{mt} is

$$g_{mt}(\mathbf{u}^*) \equiv \frac{\partial}{\partial \lambda_{mt}} \Phi(\boldsymbol{\lambda}) = \sum_{i=1}^{I} \sum_{\tau=t-P_{im}+1}^{t} u_{im\tau}^* - C_m,$$

and the Lagrange multipliers are updated by

$$\lambda_{mt}^{k+1} = \begin{cases} \lambda_{mt}^k + \alpha^k g_{mt}(\mathbf{u}^*), & \text{if } \lambda_{mt}^k > 0 \text{ or} \\ & \text{if } \lambda_{mt}^k = 0 \text{ and } g_{mt}(\mathbf{u}^*) \ge 0, \\ \lambda_{mt}^k, & \text{if } \lambda_{mt}^k = 0 \text{ and } g_{mt}(\mathbf{u}^*) < 0, \end{cases}$$

where $\alpha^k = \dfrac{\gamma[\bar{\Phi}(\boldsymbol{\lambda}^*) - \Phi(\boldsymbol{\lambda}^k)]}{g(\boldsymbol{\lambda}^k)^T g(\boldsymbol{\lambda}^k)}$ is the step size at the kth iteration, $0 < \gamma < 2$, and $\bar{\Phi}(\boldsymbol{\lambda}^*)$ is an estimate of the optimal dual cost

The iteration step terminates if α^k is smaller than a threshold. Polyak proved that this method has a linear convergence rate [8].

III.4 Construction of a Good Feasible Schedule

The schedule derived from the dual problem is generally infeasible for the primal problem (P) because (P) is not a convex programming problem [5] and the machine capacity constraints (2.2) may not be satisfied after the relaxation. We develop a heuristic algorithm to adjust the schedule to a feasible one. We check capacity constraints in an ascending sequence of indices m and t. For each violated capacity constraint, we

(1) determine the priority factor (PF) of each type of parts:

$$PF = CR - LB, \tag{3.4}$$

where

$$CR = \frac{\text{time to due date}}{\text{residual processing time}} \text{ and}$$

$$LB = \frac{\text{residual processing time}}{\text{total required processing time}},$$

by using the current schedule; the lower the PF value, the higher the priority, and

(2) reroute excessive production flows starting from the type of the largest priority.

Step (2) is done by constructing a downstream (both in time and in machine) sequence subnetwork of the nodes corresponding to the constraint under consideration. We first determine the way that the excessive production flow should be pulled out by solving a MCLNF problem of the subnetwork and then reroute the flow by solving another MCLNF problem.

IV. NUMERICAL RESULTS

In this section, we present very preliminary testing results of our scheduling algorithm to show its feasibility. We also compare it with a heuristic algorithm that is being used by a few local manufacturing firms. The heuristic algorithm schedules production work orders basically on a first come first serve basis. When there are competitions for the same machine, the priority of a work order is determined by the priority factor defined in Eq. (3.4). We test both algorithms on the following example.

Example

Table 4.1 Production Processes

Part Type	Production Procsee
1	m1 -> m2 -> m4*
2	m1 -> m3 -> m4

* The notation "mn" refers to machine group n.

Table 4.2 Production Parameters

Machine Group		1	2	3	4
Machine Capacity		5	6	6	5
Proc. Time	Part Type 1	1	2	x	2
	Part Type 2	1	x	1	2

Three cases are created by varying demands and due dates (Table 4.3). Exhaustive search is used to determine the optimal costs for each case [6]. Test results are listed in Table 4.4. In case 1, the optimal solution is achieved by both our algorithm and the heuristic. In case 2, our algorithm yields an optimal solution while the heuristic leads to a suboptimal solution. In case 3, our algorithm leads to a near-optimal solution but much better than that of the heuristic.

Table 4.3 Three Test Cases

Case	Part Type	Final Demand	Due Date
1	1	3	6
	2	3	4
2	1	8	10
	2	6	4
3	1	8	8
	2	6	4

Table 4.4.a Comparison of Costs

Case	Optimal Cost	Dual Cost	Our Algorithm	Heuristic
1	60000.0	59998.7	60000.0	60000.0
2	160000.0	143026.0	160000.0	220000.0
3	200000.0	147205.0	240000.0	440000.0

Table 4.4.b Comparison of CPU Time (in seconds)

Case	Our Algorithm	Heuristic
1	1.05	0.16
2	1.29	0.19
3	1.37	0.20

V. CONCLUDING REMARKS

We have developed in this paper a near-optimal solution algorithm for a class of flow shop sheduling problems. The algorithm combines optimization techniques of decomposition by Lagrangian relaxation, minimum cost network flow algorithm, a subgradient method and a feasibility adjustment heuristic. Though a shop of flow line structure has been used to convey the idea, it can be easily seen that the algorithm applies to more general flow shops of no setup costs. Preliminary numerical results have demonstrated that the algorithm is feasible and potentially outperforms a simple heuristic in solution optimality. We are tuning the algorithm for better solution and computational efficiency, especially the subgradient algorithm and the feasibility adjustment heuristic. Further numerical tests on realistic problems and analysis will be performed in the future to better explore the properties and potential of our algorithm.

REFERENCES

[1] Bertsekas, D. P., and P. Tseng, "Relaxation Methods for Minimum Cost Ordinary and Generalized Network Flow Problem", Operations Research, Vol. 36, No. 1, 1988, pp. 93–114.

[2] Fisher, M. L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems", Management Science, Vol. 27, No. 1, Jan. 1981, pp. 1–18.

[3] Technical Memo, Information and Administration Dept., Fu Sheng Industrial Co., San-Chung, Taiwan, 1989.

[4] Graves, S. C., "A Review of Production Scheduling", Operations Research, Vol. 29, July–August, 1981, pp. 646–675.

[5] Luh, P. B., D. J. Hoitomt, E. Max and K. R. Pattipati, "Parallel Machine Scheduling Using Lagrangian Relaxation", Proceedings of the First International Conference on Computer Integrated Manufacturing, Troy, NY, May, 1988, pp. 244–248.

[6] Syslo, M. M., N. Deo and J. Kowalik, Discrete Optimization Algorithms with PASCAL Programs, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1983.

[7] Vollmann, T. E., W. L. Berry and D. C. Whybark, Manufacturing Planning and Control Systems, 2nd ed., the Dow Jones-Irwin/APICS Series in Production Management , Homewood, IL, 1988.

[8] Polyak, B. T., "Minimization of Unsmooth Functionals", USSR Computional Math. and Math. Physics, Vol. 9, 1969, pp. 14–29.