Journal of Manufacturing Systems
Vol. 18/No. 4
1999

An Effective Optimization-Based Algorithm for Job Shop Scheduling with Fixed-Size Transfer Lots

Bin Jin and **Peter B. Luh,** Dept. of Electrical and Systems Engineering, and **L.S.Thakur,** School of Business Administration, University of Connecticut, Storrs, Connecticut

Abstract

Effective scheduling of production lots is of great importance for manufacturing medium to high-volume products that require significant setup times. Compared to traditional entire-lot production, lot splitting techniques divide a production lot into multiple smaller sublots so that each sublot can be "transferred" from one stage of operation to the next as soon as it has been completed. "Transfer lots," therefore, significantly reduce lead times and lower work-in-process (WIP) inventory. The mathematical modeling, analysis, and control of transfer lots, however, is extremely difficult. This paper presents a novel integer programming formulation with separable structure for scheduling job shops with fixed-size transfer lots. A solution methodology based on a synergistic combination of Lagrangian relaxation, backward dynamic programming (BDP), and heuristics is developed. Through explicit modeling of lot dynamics, transfer lots are handled on standard machines, machines with setups, and machines requiring all transfer lots within a production lot to be processed simultaneously. With "substates" and the derivation of DP functional equations considering transfer lot dynamics, the standard BDP is extended to solve the lot-level subproblems. The recently developed "time step reduction technique" is also used for increased efficiency. It implicitly establishes two time scales to reduce computational requirements without much loss of modeling accuracy and scheduling performance, thus enabling resolution of long-horizon problems within controllable computational requirements. The method has been implemented using object-oriented programming language C++, and numerical tests show that high-quality schedules involving transfer lots are efficiently generated to achieve on-time delivery of products with low WIP inventory.

Keywords: *Job Shop Scheduling, Transfer Lots, Optimization, Lagrangian Relaxation, Dynamic Programming*

1. Introduction

Scheduling of production lots is of great importance for manufacturing medium to high-volume products with significant setups. In traditional production, a lot is *indivisible*, and individual pieces must wait for the completion of all the pieces within the lot before moving on to the next stage of operation. This often results in long lead times, low machine utilization, and high work-in-process (WIP) inventory.[1] In recent years, lot-splitting techniques have been used to divide a production lot (briefly a 'lot') into multiple smaller sublots so that each sublot can be individually "transferred" from one stage of operation to the next as soon as that sublot has been completed. Lot splitting techniques thus allow individual "transfer lots" to be concurrently processed at several production stages, which reduces manufacturing lead times, lowers WIP inventory levels, and improves product delivery times.[2-4]

Most lot splitting techniques in the literature are based on heuristics. For example, in Vickson and Alfredsson[5] two and three-machine flow shop problems with equal-sized transfer lots and a makespan objective function were solved by using a modified Johnson's algorithm where each transfer lot was treated as an independent unit. Heuristics for flow shop problems with three or more machines were presented in Trietsch and Baker.[1] A heuristic method for the integrated determination of transfer lot sizes and production schedules for a two-stage flow shop with a maximum flow time objective function was presented in Cetinkaya.[6] A simulation model for scheduling job shops with lot splitting using dispatching rules and a mean flow time objective function was presented in Jacobs and Bragg.[7] A heuristic algorithm is developed to minimize the makespan for a three-stage production process in Glass, Gupta, and Potts.[8] These heuristic approaches usually generate feasible schedules quickly and demonstrate the benefits of transfer lots toward reducing lead times and lowering WIP inventory. However, it is difficult to evaluate the quality of the schedules generated, and these heuristics do not provide a systematic way for iterative improvement of the schedules. Recently, an optimization-based method has shown promise in scheduling transfer lots on standard machines without setups.[9] The method, however, cannot handle problems with long planning horizons or machine setups. There are also situations where all transfer lots within a production lot are required to be processed simultaneously. An example is the outsourcing of the entire lot, say, for heat treatment. Extension of the method is therefore needed to solve practical problems involving transfer lots.

Building on the above results of Liu and Luh,[9] a novel integer programming formulation with separable structure for scheduling job shops with fixed-size transfer lots is presented in Section 2. In the formulation, transfer lots are handled on standard machines, machines with setups, and machines requiring all transfer lots within a production lot to be processed simultaneously. The formulation is "separable" in the sense that the objective function and all "coupling" machine capacity constraints are additive in terms of the basic decision variables at the lot level. A solution methodology based on a synergistic combination of Lagrangian relaxation (LR), backward dynamic programming (BDP), and heuristics is developed in Section 3. Through the explicit modeling of lot dynamics, the introduction of "substates," and the derivation of dynamic programming equations considering transfer lot dynamics, the standard BDP is extended to solve lot-level subproblems within the Lagrangian relaxation framework. The recently developed "time step reduction technique" is also incorporated. It implicitly establishes two time scales to reduce computational requirements without much loss of modeling accuracy and scheduling performance, which enables the resolution of long-horizon problems within reasonable computational requirements. Numerical testing results presented in Section 4 show that high-quality schedules with transfer lots are generated in a timely fashion for on-time delivery with low WIP inventory.

2. Problem Formulation

The following formulation for scheduling job shops with transfer lots is built on the previous work presented in Luh and Hoitomt[10] and Liu and Luh.[9] Instead of treating individual transfer lots as independent scheduling units as in Vickson and Alfredsson,[5] only each production lot is treated as a scheduling unit, with operation beginning and completion times as decision variables. The key is to properly describe transfer lot dynamics using production lot variables only. The preliminaries that lead to the formulation are presented first, using the symbols given in Appendix C.

Notation and General Description

Time Step Reduction. In view of the long planning horizon under consideration relative to the time resolution required (for example, six months vs. six minutes), the "time step reduction technique" originally developed in Luh et al.[11] is extended for scheduling transfer lots. The time horizon is divided into T "resolution increments" indexed by t, $0 \leq t \leq T-1$, and R consecutive resolution increments are aggregated into an "enumeration step" indexed by k, $0 \leq k \leq K$, with $T = R \times K$. For example, a 500-hour horizon can be divided into 500 one-hour resolution increments, and 50 10-hour enumeration steps by aggregating 10 one-hour resolution increments in an enumeration step. Then an operation requiring, say, 16 hours on a machine is represented as occupying a full 10-hour enumeration step and 60% of the next enumeration step. Thus, by using fractional but quantitized machine utilization, multiple "short" operations are allowed to "share" a machine within an enumeration step, and a part with several short operations is allowed to flow through the machines within a single enumeration step. Most input data are specified in terms of resolution increments except when stated otherwise. Since the complexity of the method depends significantly on the number of enumeration steps, this technique reduces computational effort in solving the "dual problem" through appropriate selection of the number of resolution increments R within an enumeration step.

Machines. In a job shop, machines may have different processing capabilities, for example, processing speed or setup requirements. Machines with identical processing capability from the scheduling point of view are grouped as a "machine type," and all the machine types form a set denoted by H.

Lots and Transfer Lots. Suppose there are L production lots, indexed by $l = 0, 1, ..., L-1$, each consisting of a number of products of the same type. For simplicity of presentation, a *production lot* will be referred to as a *lot* hereafter when there is no confusion. Different lots may have different product types, due dates, or arrival dates. For feasibility, a long-enough planning horizon T is selected that is sufficient for the completion of all L lots.

A production lot, say lot l, consists of N_l fixed and equal-sized transfer lots. It has to go through a sequence of operations, indexed by $j = 0, 1, ..., J_l - 1$, according to a specified process plan. Operation j of lot l, denoted as (l, j), has to be performed by a machine belonging to an *eligible machine type* $h \in H_{lj}$, $h = 0, 1, ..., |H_{lj}| - 1$. Once started, the entire lot (that is, all the transfer lots of the production lot) must be finished on the machine before anything else can be processed by the machine. This assumption applies to various situations, for example, when setup costs are significant or when mixed transfer lots at a machine are difficult to manage because of operator or shop-floor tracking system requirements. Let t_{ljh} denote the processing time in resolution increments per transfer lot for operation (l, j) on a machine type h. Let s_{lj} represent the required "time-out" in resolution increments between (l, j) and $(l, j+1)$, representing processes not explicitly modeled in the problem formulation, such as transfer time, cooling down time, or curing time. It is assumed that the number of transfer lots, N_l, the transfer lot processing time, t_{ljh}, and the time out, s_{lj}, are given.

Setups. If operation (l, j) has a setup requirement on a machine, it cannot be started until the machine has been set up. Assume that the setup time t^s_{ljh} in resolution increments for operation (l, j) on machine type h is known and is sequence independent (that is, t^s_{ljh} does not depend on what was processed earlier on that machine). Then, once the machine is set up for operation (l, j), a transfer lot can be started on the machine as soon as it has arrived from its predecessor operation $(l, j-1)$ and the machine has finished the predecessor transfer lot if it exists. Also, as stated earlier, the machine cannot process anything else until all the transfer lots in lot l are finished.

Decision Variables. The beginning and completion times of operation (l, j) in resolution increments are denoted by b_{lj} and c_{lj}, respectively, and are the major decision variables. To ensure schedule acceptability, b_{lj} is constrained by its given earliest beginning time b^e_{lj} and the latest beginning time b^l_{lj}, that is, $b^e_{lj} \leq b_{lj} \leq b^l_{lj}$; similarly, c_{lj} satisfies $c^e_{lj} \leq c_{lj} \leq c^l_{lj}$. These earliest and latest beginning and completion times are determined based on factors such as the arrivals of raw materials, the desire to minimize WIP inventory, and due dates promised to customers.

Machine Capacity Constraints and Lot Dynamics

Machine Capacity Constraints. The number of machines available per type at each resolution increment is a given integer. The average number of type h machines available at *enumeration step* k, denoted as M_{kh}, is thus a quantitized fraction. The machine capacity constraints state that the total number of lots being processed should not exceed the number of machines available at each time period:

$$\sum_{l=0}^{L-1} \sum_{j=0}^{J_l-1} \delta_{ljkh} \leq M_{kh} \tag{1}$$

In the above, δ_{ljkh} is the fraction of time that operation (l, j) of production lot l is assigned to machine type h at *enumeration step* k. It is also assumed that the capacity of machine types where all transfer lots of a lot must be simultaneously processed is large enough to accommodate the entire lot.

The dynamics of transfer lots is described through operation precedence constraints, processing time requirements, and the setup requirements as follows.

Operation Precedence Constraints. Assume that a machine of type h has been set up for $(l, j+1)$ of lot l so that this operation can be started. The operation precedence constraints require that operation $(l, j+1)$ cannot be started until the predecessor operation (l, j) of the *first* transfer lot has been completed, that is,

$$b_{lj} + t_{ljh} + s_{lj} \leq b_{l,j+1} \tag{2}$$

where s_{lj} is any required "time out." If the operation (l, j) is performed on machines that require simultaneous processing of all transfer lots within a lot, then the constraints become that $(l, j+1)$ cannot be started until the *last* transfer lot has completed operation (l, j), that is,

$$c_{lj} + s_{lj} + 1 \leq b_{l,j+1} \tag{3}$$

The presence of "1" in (3) is due to the convention that when an operation *begins* in a period, it starts at the beginning of that period; however, when it *ends* in a period, it finishes at the end of that period, thus occupying the entire period in both cases. This convention is often followed in practice.

Operation Processing Time Requirements. Each operation beginning time b_{lj} may be associated with multiple completion times c_{lj} since the completion time depends not only on the operation beginning time and the transfer lot processing time, t_{ljh}, but also on the lengths of *intermittent idling times.* Despite the availability of a machine, intermittent idling between two transfer lots may exist because the next transfer lot may still be in processing at the previous stage if the processing time there is longer than the current one.[1,9] The lengths of intermittent idling times depend on several factors; nevertheless, it is clear that if there is no intermittent idling between b_{lj} and c_{lj}, then $c_{lj} = b_{lj} + N_l \times t_{ljh} - 1$; otherwise, $c_{lj} = c_{l,j-1} + s_{l,j-1} + t_{ljh}$. Consequently, the operation completion time, c_{lj}, is given by the following:

$$c_{lj} = \max(b_{lj} + N_l \times t_{ljh} - 1,\ c_{l,j-1} + s_{l,j-1} + t_{ljh}) \qquad (4)$$

The significance of (4) is that, although the movement of individual transfer lots is not explicitly modeled, production lot beginning and completion times can be accurately described by (2) and (4).

If operation (l, j) is performed on a machine that processes all transfer lots within a lot simultaneously, then all the transfer lots have the same beginning and completion times. The completion time thus depends only on the beginning time, b_{lj}, and the transfer lot processing time, t_{ljh}, that is,

$$c_{lj} = b_{lj} + t_{ljh} - 1 \qquad (5)$$

Setup Requirements. For an operation (l, j) requiring setups, although the first transfer lot has to wait for the completion of its predecessor operation (l, j−1), the machine's setup can be started earlier. The actual setup beginning time for operation (l, j), denoted by b'_{lj}, is

$$b'_{lj} = b_{lj} - t^s_{ljh} \qquad (6)$$

assuming that the machine is available for setup at b'_{lj}. Therefore, once operation beginning time b_{lj} is known, the setup beginning time can be readily computed, and setups can be embedded within lot dynamics without introducing additional variables.

Objective Function

The objective of scheduling is to ensure on-time product delivery with low WIP inventory. This is represented by minimizing the sum of weighted quadratic penalties for violating lot due dates and for releasing raw materials too early:

$$J \equiv \sum_l \left(w_l T_l^2 + \beta_l E_l^2\right) \qquad (7)$$

In the above, T_l is the tardiness of lot l defined as the time the lot completion time, c_l (completion time of the *last* operation of the *last* transfer lot), exceeds the given lot due date, d_l, in enumeration steps, that is,

$$T_l \equiv \max\left(0, \left\lfloor \frac{c_l}{R} \right\rfloor - \left\lfloor \frac{d_l}{R} \right\rfloor\right)$$

The lot earliness, E_l, is similarly defined as the excess of the lot's earliest beginning time, $\bar{b}_l$, over the lot's scheduled beginning time, b_l (the beginning time of the *first* operation of the *first* transfer lot), that is,

$$E_l \equiv \max\left(0, \left\lfloor \frac{\bar{b}_l}{R} \right\rfloor - \left\lfloor \frac{b_l}{R} \right\rfloor\right)$$

The parameters w_l and β_l are weights associated with the earliness and tardiness penalties of lot l. The above penalties define a time window in which the lot can be scheduled without penalty.

The key decision variables are operation beginning times $\{b_{lj}\}$ and completion times $\{c_{lj}\}$ for individual production lots. Once these variables are determined, transfer lot beginning times and completion times can be derived as presented in Appendix A.

3. Solution Methodology

Lagrangian relaxation (LR) is a mathematical programming technique for constrained optimization. Similar to the pricing concept of a market economy, the method replaces "hard" coupling constraints (that is, machine capacity constraints in this study) by the payment of certain "prices" (Lagrange multipliers) based on the "demand" for a machine for the use of that machine at each time unit. The original problem can thus be decomposed into many smaller and easier lot-level subproblems. Backward dynamic programming (BDP) is then used to solve these lot-level subproblems where other constraints are enforced. The multipliers are then adjusted after these subproblems are solved, based on the degrees of constraint violation, following again the market economy mechanism. Subproblems are then re-solved based on the new set of multipliers, and the process repeats. In mathematical terms, the "dual function" is maximized in this multiplier updating process, where the values of the dual function are lower bounds to the optimal feasible cost. Since coupling constraints have been relaxed by the multipliers, the solutions of individual subproblems, when put together, may not constitute a feasible schedule. A simple heuristic is therefore used toward the end of this multiplier updating process to provide feasible schedules satisfying all constraints. The quality of the feasible schedules can be quantitatively evaluated by comparing their costs with the largest lower bound provided by the dual function. The development of the BDP technique is complicated and will be one of the major topics covered in this section.

Lagrangian Relaxation Framework

Machine capacity constraints (1) are first "relaxed" by using Lagrange multipliers $\{\pi_{kh}\}$ in *enumeration steps*, and the Lagrangian is formed as:

$$J^+ \equiv \sum_{l=0}^{L-1}\left(w_l T_l^2 + \beta_l E_l^2\right) + \sum_k \sum_h \pi_{kh} \left(\sum_{l=0}^{L-1}\sum_{j=0}^{J_l-1} \delta_{ljkh} + m_{kh} - M_{kh}\right) \quad (8)$$

where m_{kh} is a non-negative slack variable satisfying $0 \le m_{kh} \le M_{kh}$. With the multipliers given, the "relaxed problem" is to minimize the Lagrangian J^+ subject to operation precedence constraints, processing time requirements, and setup requirements (2) to (6). After regrouping relevant terms within J^+, the problem is decomposed into individually solvable *lot subproblems* as follows:

$$\min_{\{b_{lj},\, c_{lj}\}} L_l, \text{ with } L_l \equiv w_l T_l^2 + \beta_l E_l^2 + \sum_{j=0}^{J_l-1} L_{lj}\left(b_{lj}, c_{lj}\right), \text{ and}$$
$$L_{lj}\left(b_{lj}, c_{lj}\right) \equiv \sum_{t=b_{lj}-t_{ljh}^s}^{c_{lj}} \tilde{\pi}_{th} \quad (9)$$

To simplify the derivation, $\tilde{\pi}_{th}$ is introduced to represent the multiplier for machine type h at resolution increment t, that is,

$$\tilde{\pi}_{th} \equiv \frac{\pi_{kh}}{R} \text{ with } k \equiv \left\lfloor \frac{t}{R} \right\rfloor$$

The subproblems $\{L_l\}$ are subject to (2) to (6). The decision variables are operation beginning times $\{b_{lj}\}$ and completion times $\{c_{lj}\}$ of lot l.

Since L_{lj} (b_{lj}, c_{lj}) is the cost for using a type h machine between times b_{lj} and c_{lj}, the lot subproblem thus reflects the balance between the machine utilization cost and tardiness as well as earliness penalties.

Backward Dynamic Programming (BDP) for Lot Subproblems: Preliminaries

In view of the complications caused by the existence of multiple completion times associated with a given beginning time, the BDP algorithm developed in Luh et al.[11] must be extended to solve lot subproblems. In the following, the generic DP equations are first presented. Several key parameters are then determined, including the number of multiple operation completion times associated with each beginning time, the earliest beginning and completion times, and the latest beginning and completion times. These parameters help reduce BDP computational requirements. In addition, "substates" are introduced to efficiently carry out the DP procedure.

Backward Dynamic Programming Equations

Each lot subproblem has a number of DP stages, where each stage corresponds to an operation. The BDP algorithm starts with the last stage and moves backward in time. The states for a stage correspond to possible beginning times in Luh et al.[11] In view of the multiple completion times associated with a given beginning time, a state in this study is represented by a pair of operation beginning and completion times (b_{lj}, c_{lj}). The cumulative cost at (b_{lj}, c_{lj}), denoted by V_{lj} (b_{lj}, c_{lj}), is obtained as the sum of the stagewise cost L_{lj} (b_{lj}, c_{lj}) (plus tardiness penalty for the last stage and earliness penalty for the first stage) and the minimum cumulative cost of a reachable state $(b_{l,j+1}, c_{l,j+1})$ at the successor stage.

To be more specific, the BDP procedure starts with the last stage having the following terminal cost:

$$V_{l,J_l-1}\left(b_{l,J_l-1}, c_{l,J_l-1}\right) \equiv w_l T_l^2 + L_{l,J_l-1}\left(b_{l,J_l-1}, c_{l,J_l-1}\right) \quad (10)$$

The cumulative cost when moving backward to the predecessor stage is then obtained recursively according to the following BDP equation subject to constraints (2) to (6):

$$\begin{aligned} V_{lj}(b_{lj}, c_{lj}) &\equiv \min_{\{(b_{l,j+1},\, c_{l,j+1}) \in S_{l,j+1}(b_{lj}, c_{lj})\}} \{L_{lj}(b_{lj}, c_{lj}) + V_{l,j+1}(b_{l,j+1}, c_{l,j+1})\} \\ &= L_{lj}(b_{lj}, c_{lj}) + \min_{\{(b_{l,j+1},\, c_{l,j+1}) \in S_{l,j+1}(b_{lj}, c_{lj})\}} \left\{V_{l,j+1}(b_{l,j+1}, c_{l,j+1})\right\}, \\ & 1 \le j < J_l - 1 \end{aligned} \tag{11}$$

In the above, $S_{l,j+1}$ (b_{lj}, c_{lj}) is the set containing all allowable $\{(b_{l,j+1}, c_{l,j+1})\}$ satisfying (2) to (6) for the given (b_{lj}, c_{lj}) and is determined based on possible state transitions.

The equation for the first stage is given by the following:

$$V_{l0}(b_{l0}, c_{l0}) = \beta_l E_l^2 + L_{l0}(b_{l0}, c_{l0}) + \min_{\{(b_{l1},\, c_{l1}) \in S_{l1}(b_{l0}, c_{l0})\}} \left\{V_{l1}(b_{l1}, c_{l1})\right\} \tag{12}$$

Let L_l^* represent the minimal lot subproblem cost for lot l. The minimum L_l^* can then be obtained as the minimal cumulative cost at the first stage, that is,

$$L_l^* \equiv \min_{\{b_{l0}, c_{l0}\}} \left\{V_{l0}(b_{l0}, c_{l0})\right\} \tag{13}$$

Finally, the optimal beginning times and completion times can be obtained by tracing forward along the stages.

Multiple Completion Times Associated with a Beginning Time

For a given operation beginning time, the operation completion time depends on the lengths of intermittent idling times between transfer lots. As a result, there maybe multiple completion times associated with a given beginning time. A detailed analysis of this complicated phenomenon can be found in Liu and Luh.[9] To deal with transfer lots effectively, a forward procedure is introduced here to determine these multiple completion times. For operation (l, j), the number of multiple completion times associated with a beginning time, denoted by N_{lj}^c+1, is determined by considering the machine type being used, its processing time, and the processing times of its previous operations, as given in Appendix B.

Earliest and Latest Operation Beginning and Completion Times

For a given planning horizon, in view of constraints (2) to (6), the beginning time as well as the completion time of each operation must have the earliest (least) value and the latest (largest) value, respectively. Moreover, these parameters, the earliest and latest operation beginning and completion times, are helpful in effectively limiting the computational effort in BDP. Thus, the *earliest* beginning and completion times are determined to ensure that every operation can be started and completed as early as possible and that these earliest times satisfy constraints (2) to (6) within the given time horizon. The earliest beginning and completion times of the current operation can be obtained recursively by proceeding forward from its predecessor operation while enforcing constraints (2) to (6).

Similarly, computed values of the *latest* beginning and completion times ensure that the latest completion times of their successor operations are still within the planning horizon T and they satisfy constraints (2) to (6). The latest beginning and completion time of the current operation can be calculated recursively by proceeding backward from its successor operation while enforcing constraints (2) to (6).

Backward Dynamic Programming Structure

Similar to Luh et al.,[11] the DP *stages* correspond to operations and DP *states* to the possible operation beginning times. The numbers of DP states are determined by the earliest and latest beginning times. Since transfer lots lead to multiple completion times for a beginning time , DP substates are introduced to consider these multiple values.

Let N_{lj}^c+1 denote to the number of multiple completion times associated with a beginning time for a stage (l, j) and a DP substate be a pair of the feasible operation beginning time and one of its associated completion times. The substate in a state is indexed by 0, 1, ..., N_{lj}^c; substate 0, substate 1, ..., and substate N_{lj}^c correspond to the pairs of the beginning time and the completion time between which there is zero unit, one unit, ..., and N_{lj}^c units intermittent idling time, respectively. For simplicity (without loss of generality), substate 0 in a state is considered to coincide with that state itself; that is, it is

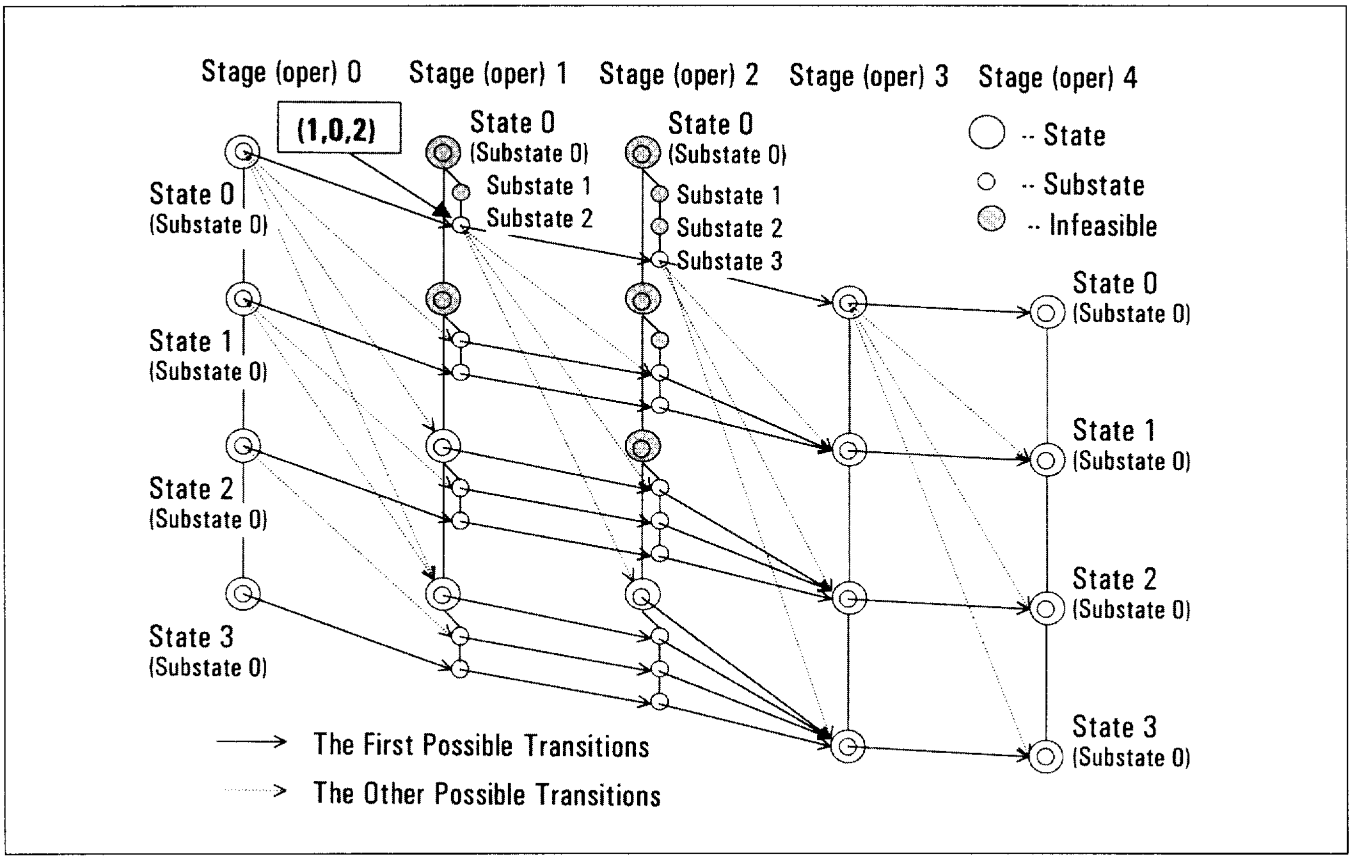

Figure 1
DP Stages, States, Substates, and Transitions

not necessary to give the corresponding completion time explicitly because there is no (zero) intermittent idling.

Figure 1 shows a sample of stages, states, substates, and state transitions, where a substate in a state is represented in the format: (stage index, state index, substate index) and state transitions follow constraints (2) to (6). For example, (1, 0, 2) represents the second substate in state 0 at stage 1. Based on this structure, the BDP procedure for solving a subproblem is now presented.

Backward Dynamic Programming (BDP) for Lot Subproblems

Unlike FDP, which moves from the first to last stage, in BDP the cumulative cost, given by (11), is computed from the last state b^l_{lj} to the first state b^e_{lj} (indexed by $N^b_{lj} - 1$ to 0), and for each state, from the last substate to the first substate (indexed by N^c_{lj} to 0). For efficiency, determining the set $S_{l,j+1}$ (b_{lj}, c_{lj}) of feasible successor states is critical.

Set of All Feasible States at Successor Stage $S_{l,j+1}$ (b_{lj}, c_{lj})

Let parameters t_b and t_c denote the earliest possible beginning and completion times of $(b_{l,j+1}, c_{l,j+1}) \in S_{l,j+1}$ (b_{lj}, c_{lj}), respectively. In view of the constraints (2) to (6), the times t_b and t_c can be calculated as follows. If the operation (l, j+1) is processed on standard machines and machines with setup, then

$$t_b = b_{lj} + t_{ljh} + s_{lj},\ t_c = \max(t_b + N_l \times t_{l,j+1,h} - 1,\ c_{lj} + s_{lj} + t_{l,j+1,h}) \tag{14}$$

however, if all transfer lots are required to be processed simultaneously, then

$$t_b = c_{lj} + s_{lj} + 1,\ t_c = t_b + t_{l,j+1,h} - 1 \tag{15}$$

For the given (b_{lj}, c_{lj}), considering the operation processing time requirements (4) and (5), the completion time $c_{l,j+1}$ can be uniquely determined from each possible beginning time $b_{l,j+1}$. Since $(b_{l,j+1},$

$c_{l,j+1}) \in S_{l,j+1}(b_{lj}, c_{lj})$ may have intermittent idling times, the set $S_{l,j+1}(b_{lj}, c_{lj})$ can be decomposed into two disjoint subsets—with and without idling times—as follows:

$$S_{l,j+1}\left(b_{lj}, c_{lj}\right) \equiv S^1_{l,j+1}\left(b_{lj}, c_{lj}\right) \cup S^2_{l,j+1}\left(b_{lj}\right) \tag{16}$$

where

$$S^1_{l,j+1}(b_{lj}, c_{lj}) \equiv \{(b_{l,j+1}, c_{l,j+1}): t_b \leq b_{l,j+1} \leq t_b + N^c_{l,j+1},\ c_{l,j+1} = t_c\} \tag{17}$$

and

$$S^2_{l,j+1}(b_{lj}) \equiv \{b_{l,j+1}: t_b + N^c_{l,j+1} + 1 \leq b_{l,j+1} \leq b^l_{l,j+1}\} \tag{18}$$

In the first set defined by (17), every element has intermittent idling times, and there are a total of $N^c_{l,j+1}$ elements with the same $c_{l,j+1}$ (equal to t_c). In the second set defined by (18), each element is a substate with index 0 because of no intermittent idling time and is represented by the state itself. For example, as shown in *Figure 1*, for the substate (1, 0, 2), its (t_b, t_c) is represented by the substate (2, 0, 3); the transition of the substate (1, 0, 2) is represented by $S^1_{l2}(0,2) = \{(2, 1, 2), (2, 2, 1)\}$ and $S^2_{l2}(0) = \{(2, 2, 0), (2, 3, 0), ..., (2, N^b_{l2} - 1, 0)\}$, where N^b_{l2} represents the number of total states in stage 2.

Stagewise Minimum Cumulative Cost of the Successor Stage

In computing the cost $V_{l,j}(b_{lj}, c_{lj})$ by (11), the major computational work is to find the minimum cumulative cost among all elements in $S_{l,j+1}(b_{lj}, c_{lj})$. The following computation-critical equations are developed to obtain the minimum with the least amount of computation. Let $F_{l,j+1}(t_b, t_c)$ denote the minimum cumulative cost of the successor stage $(l, j+1)$ within time (t_b, t_c) among all possible $(b_{l,j+1}, c_{l,j+1})$ ($S_{l,j+1}(b_{lj}, c_{lj})$, that is,

$$F_{l,j+1}\left(t_b, t_c\right) \equiv \min_{\left\{b_{l,j+1} \geq t_b,\, c_{l,j+1} = \max\left(t_c,\, b_{l,j+1} + N_l t_{l,j+1,h} - 1\right)\right\}} V_{l,j+1}\left(b_{l,j+1}, c_{l,j+1}\right) \tag{19}$$

In view of (15) and (16), $F_{l,j+1}(t_b, t_c)$ can be recursively obtained by comparing the cost $V_{l,j+1}(t_b, t_c)$ at time t_b and the minimum of the costs $(F_{l,j+1}(t_b + 1, t_c)$, $F_{l,j+1}(b_{lj}, c_{lj})$, $F_{l,j+1}(t_b + 1, t_c + 1)$ at time $t_b + 1$. This can be expressed as follows. If the operation $(l, j+1)$ is processed on the standard machines or machines with setups, then

$$F_{l,j+1}(t_b,t_c) = \min\left[V_{l,j+1}(t_b,t_c), F_{l,j+1}(t_b + 1, t_c)\right], \quad \text{if } t_c > t_b + N_l \times t_{ljh} - 1; \tag{20}$$

$$F_{l,j+1}(t_b,t_c) = \min\left[V_{l,j+1}(t_b,t_c), F_{l,j+1}(t_b + 1, t_c + 1)\right], \quad \text{if } t_c = t_b + N_l \times t_{ljh} - 1 \tag{21}$$

if all transfer lots are required to be processed simultaneously, then

$$F_{l,j+1}(t_b,t_c) = \min\left[V_{l,j+1}(t_b,t_c), F_{l,j+1}(t_b + 1, t_c + 1)\right] \tag{22}$$

The significance of (20) to (22) is that in most situations only one step comparison is needed to obtain $F_{l,j+1}(t_b, t_c)$. As a result, the cumulative cost in (11) can be rewritten as follows:

$$V_{lj}(b_{lj}, c_{lj}) = L_{lj}(b_{lj}, c_{lj}) + F_{l,j+1}(t_b t_c) \tag{23}$$

The computational complexity of the above BDP is, $O(\sum_j N^c_{lj} T)$ the same as that of FDP analyzed in Liu and Luh.[9]

Required Setups and Simultaneous Processing of All Transfer Lots

The setup times t^s_{ljh} for transfer lots are not explicitly expressed in the above BDP equations; they are considered in the computation of stagewise-state cost $L_{l,j}(b_{lj}, c_{lj})$ by (9). Thus, setup requirements have little effect on BDP computational efficiency.

However, as given by (20) to (22), BDP equations are affected by the machines requiring simultaneous processing of all transfer lots in a lot. Simultaneous processing is required, for example, in a heat treatment operation that is subcontracted out. Here all production lot units—that is, all transfer lots—are sent to the subcontractor together for the heat treatment operation. But for operations processed on these machines, substates are only with index 0, not adding much computational burden in the BDP procedure.

Slack Variable Subproblems and the Dual Problem

Slack Variable Subproblems

Little effort is needed to solve the slack variable subproblems because these subproblem solutions only require summations of the multipliers, as follows:

$$\min_{m_{kh}} L_s, \text{ with } L_s = \sum_{k,h} \pi_{kh} m_{kh} \quad (24)$$

The Dual Problem

With the optimal costs of lot subproblems and slack variable subproblems given by $\{L_l^*\}$ and L_s^*, respectively, the high-level dual problem, denoted by D, is obtained as follows:

$$\max_{\{\pi_{kh}\}} D, \text{ with } D \equiv \sum_l L_l^* + L_s^* - \sum_{k,h} \pi_{kh} M_{kh} \quad (25)$$

Since the dual function D is concave, piecewise linear, and consists of many facets, the subgradient method is commonly used to solve it; however, this method suffers from slow convergence. To overcome this, instead of solving *all* subproblems for multipliers updates, the Interleaved Sub-Gradient (ISG) method was suggested that updates multipliers after solving *each* subproblem. Furthermore, since the dual function approaches a smooth function as the problem size increases, the Conjugate Gradient (CG) methods have more attractive convergence properties for such problems. Therefore, the newly developed Interleaved Conjugate Gradient (ICG) method that incorporates the "interleave" concept with the CG method can provide faster convergence.[12] It is used to update the multipliers in this study.

As presented in Luh et al.,[11] a rough estimate of the number of multipliers when there are K enumeration steps is π_{kh}:K × |H|. With the "time step reduction" technique, there is no need to have a multiplier for each of the resolution steps T, thus reducing the computational complexity for solving the high-level dual problems significantly.

Heuristics

The computation of subproblem solutions and multiplier updates is stopped after a fixed amount of computation time or a fixed number of iterations, where an iteration consists of solving all the subproblems once. Since machine capacity constraints have been relaxed, solutions of subproblems, when put together, generally do not constitute a feasible schedule. A simple heuristic procedure is usually used to adjust subproblem solutions to form a feasible schedule. The heuristics developed for transfer lots are based on the version developed by Luh and Hoitomt[10] and Luh et al.[11]

The heuristics start with the solutions of the DP subproblems that give the actual operation beginning times and completion times. Since the setup for an operation is always immediately followed by the start of that operation on the first transfer lot, this setup is scheduled based on the availability of the desired machine and the start time of the current operation of the first transfer lot. For the first transfer lot in a production lot, its current operation can be started after the completion of its predecessor operation. For other individual transfer lots in the same lot, the current operation for a transfer lot can be processed as soon as the predecessor operation of this transfer lot and the current operation of the predecessor transfer lot are complete, together with the consideration of machine availability. For the operation processed on the machine requiring simultaneous processing of all transfer lots in the same lot, since such machine capacity is treated as large, this operation can begin after the completion of the predecessor operation of the last transfer lot.

For simplicity of presentation, details of the situation are not given where multiple machine types can do a given operation; the BDP can be easily extended to this situation by considering multiple stages for an operation, one for each applicable machine type. The quality of a schedule obtained is quantitatively evaluated by its relative duality gap, which is the relative difference between the feasible schedule cost, J, and its lower bound, the dual value D; that is, Duality Gap ≡ (J − D)/D × 100%. The stopping criterion for the solution procedure may be to obtain a given duality gap within an acceptable range.

Numerical Testing Results

The current algorithm that combines BDP and ICG within the LR framework has been implemented using the object-oriented programming language C++, and extensive initial testing has been performed on a Sun Sparc 10 workstation. Four test cases are presented below to evaluate the performance of the method developed. The first case shows that using transfer lots improves the scheduling performance greatly. Case 2 illustrates that the transfer lots can be scheduled effectively on various types of machines: standard machines, machines with setups, and machines requiring all transfer lots in the same lot to be processed simultaneously. This increases the applicability of the model to a substan-

Table 1a
Data and Results for Case 1: Transfer Lots

Lot l (N_l)	Operation j	Machine h	t_{ljh}	d_l	w_l	Schedule 1.1 (b_{lj}, c_{lj})	Schedule 1.2 (b_{lj}, c_{lj})
0 (5)	0	M0	2			(12, 21)	(6, 15)
	1	M1	1			(14, 22)	(16, 20)
	2	M2	2	1	1	(15, 24)	(21, 30)
1 (2)	0	M0	3			(0, 5)	(0, 5)
	1	M1	1			(3, 6)	(6, 7)
	2	M2	2	0	1	(5, 8)	(8, 11)
2 (2)	0	M1	1			(0, 1)	(0, 1)
	1	M2	2			(1, 4)	(2, 5)
	2	M0	3	1	1	(6, 11)	(16, 21)

Table 1b
Scheduling Performance for Case 1: Transfer Lots

Schedule	Makespan	Avg. Lead Time	Avg. WIP Inventory	Avg. Machine Utilization	Avg. Tardiness
Schedule 1.1: With Transfer Lots	25	6.4	0.26	65.3%	15
Schedule 1.2: Without Transfer Lots	31	16	0.52	52.7%	21

tially larger set of realistic environments. Case 3 and Case 4 demonstrate the capability of the method developed for scheduling real problems with different sizes (number of lots, parts, transfer lots, and time horizon). All cases assume that all machines are available throughout the planning horizon, starting from period zero. Using heuristics, feasible planning horizons are initially generated based on machine availability and lot processing time requirements. For the first three cases, the enumeration step is equal to the resolution time unit (that is, R = 1). In addition to the duality gap, the following practical metrics, used by various industries, are also applied to evaluate the scheduling performance.

The metrics are as follows:

$$\text{Makespan} = \max_{l,n} c^{n}_{l,J_l-1} - \min_{l,n} b^{n}_{l0} + 1$$

Lead time of the nth transfer lot in lot $l = c^{n}_{l,J_l-1} - b^{n}_{l0} + 1$;

WIP inventory of the nth transfer lot in lot L = (lead time of the nth transfer lot in lot l) / makespan;

Machine utilization of machine h

$$= \left(\sum_{l} \sum_{j} \left(t_{1jh} + t^{s}_{1jh} \right) \right) \Big/ \text{available time in makespan}$$

and Tardiness (delivery delay) of the nth transfer lot in lot l = max $(0, c^{n}_{l,J_{l-1}} + 1 - d_l)$. In the above, b^{n}_{lj} and c^{n}_{lj} are the beginning time and completion time of the operation (l, j) of the nth transfer lot, respectively.

In addition, a comparison of the current algorithm with the common dispatching rules used in practice has been suggested by the journal editor. The resulting schedules are being computed and will be presented for examination by the readers via the Internet at http://www.sme.org.

Case 1 (Schedules With and Without Transfer Lots)

This case is to show by a small example that, as expected, scheduling with transfer lots can indeed improve the scheduling performance greatly. There are three lots to be scheduled on three machine types, with one machine of each type. There are five parts in lot 0 and two in lots 1 and 2. The data are given in *Table 1a*. The planning horizon is 40 time units (T = 40).

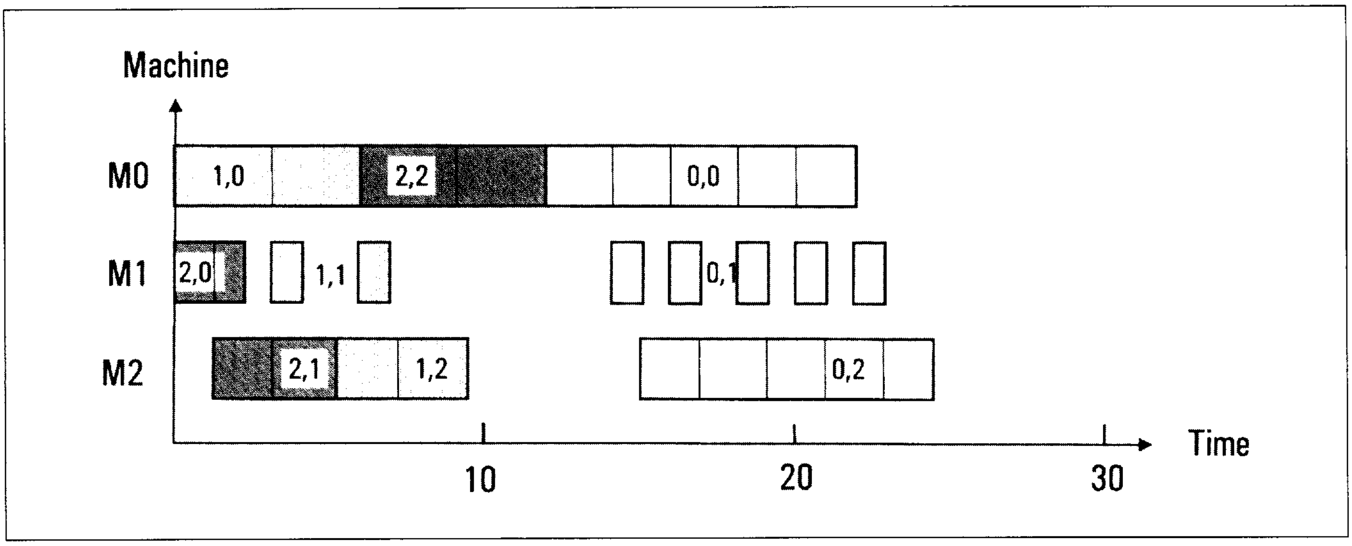

Figure 2a
Gantt Chart of Schedule 1.1: Scheduling with Transfer Lots

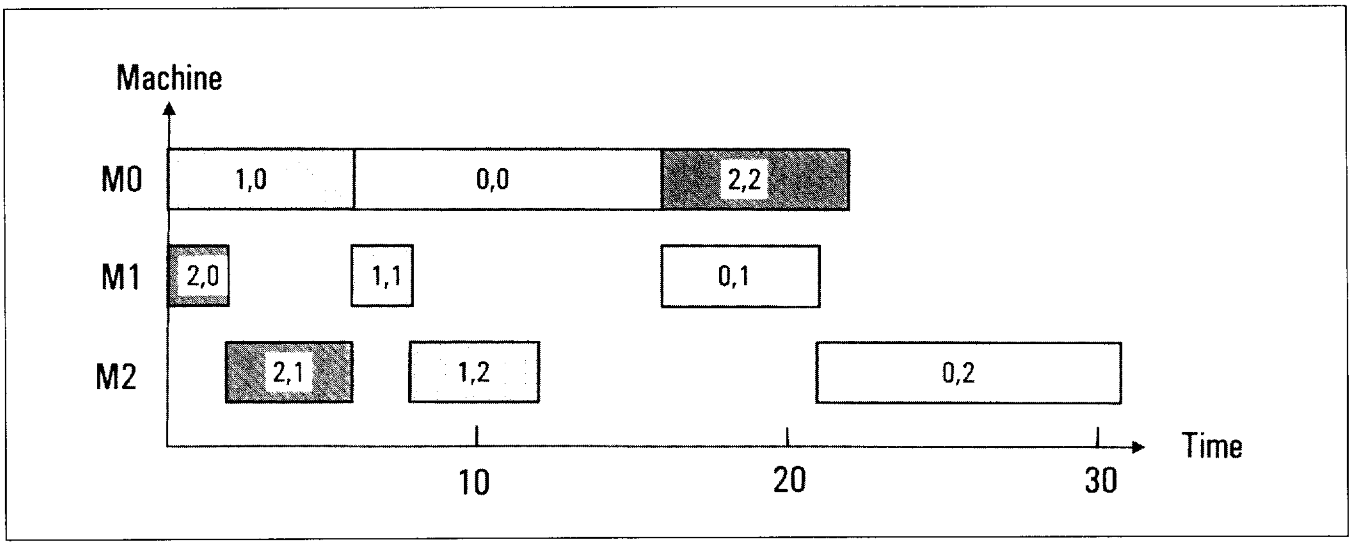

Figure 2b
Gantt Chart of Schedule 1.2: Scheduling Without Transfer Lots

The problem is first solved with transfer lots of size one. Each part is, thus, treated as a transfer lot. The feasible schedule, Schedule 1.1, is generated with a cost of 693 and a lower bound of 693 in CPU time of less than one second. Thus, the solution found is the optimal solution. The operation beginning times and completion times of Schedule 1.1 are also shown in *Table 1a*, with the Gantt chart given in *Figure 2a*.

Then the same problem is solved assuming all lots are indivisible (that is, without using transfer lots). The feasible schedule, Schedule 1.2, has a cost of 1362 with a lower bound of 1361.87 and is obtained in less than one second CPU time. The resulting operation beginning times and completion times are also presented in *Table 1a*. The Gantt chart of Schedule 1.2 is shown in *Figure 2b*.

Table 1b gives the metrics for the scheduling performance of both Schedule 1.1 and 1.2. Comparing with the schedule obtained without considering transfer lots (Schedule 1.2), transfer lots have significantly improved the average lead time, average WIP inventory, average machine utilization, and average delivery delay time. These imply that the use of

Table 2
Data and Results for Case 2: Various Machine Types

Lot l (N_l)	Operation j	Machine h	t_{ljh}	t^s_{ljh}	s_{lj}	a_l	d_l	w_l	Schedule 2 (b_{lj}, c_{lj})
0 (4)	0	M0	3	-	1	0			(6, 17)
	1	M2	2	4	-				(12, 20)
	2	M4	2	-	-				(21, 22)
	3	M1	1	-	-		0	3	(23, 26)
1 (2)	0	M1	2	-	-	0			(0, 3)
	1	M2	3	-	1				(2, 7)
	2	M0	2	2	-		1	1	(20, 23)
	3	M4	2	-	-				(24, 25)
2 (3)	0	M0	1	3	1	0			(3, 5)
	1	M1	2	-	-				(5, 10)
	2	M4	2	-	1				(11, 12)
	3	M3	5	-	-		2	2	(14, 28)
3 (3)	0	M3	2	-	-	2			(2, 7)
	1	M4	2	-	-				(8, 9)
	2	M2	1	2	-				(23, 25)
	3	M1	3	-	1		8	1	(27, 35)

sublots of smaller size and the overlapping of consecutive operations results in less work in process and less product delivery delay.

Case 2 (Scheduling Transfer Lots with Various Machine Categories)

This case is to show that the method presented here can effectively schedule transfer lots on three key machine categories: standard machines, machines with setups, and machines where all transfer lots in a lot must be processed simultaneously. There are four lots to be scheduled on five machine types, each machine type with one machine. Each lot has a different due date, and lot 0 has the highest priority (weight = 3) among the four lots, while lot 2 has a higher priority (weight = 2) than the other two. Lot 3 has an arrival time of 2 units, and each lot has four operations. Some operations processed on M0 and M2 need setups. For the operation processed on M4, all transfer lots in a lot must be processed simultaneously. The detailed data about lots and operations are given in *Table 2*. "Time outs" are also considered in this case.

The feasible schedule, Schedule 2, has a cost of 4740 with a relative duality gap 3.53% and is obtained in two seconds. In *Table 2*, the resulting operation beginning times and completion times (b_{lj}, c_{lj}) are also presented. The Gantt chart of the feasible schedule is shown in *Figure 3*. The makespan, average lead time, average WIP inventory, average machine utilization, and average tardiness are 36, 21.4, 0.59, 55%, and 19.7, respectively. This once again shows that the schedules generated involving transfer lots on all three kinds of machines are of high quality.

The following two observations on the solution presented in *Figure 3* are made to illustrate how transfer lots are scheduled effectively on the machines with setups and machines requiring simultaneous processing of all transfer lots. First: the setup for operation (0,1) is started at time 8 on M2 while the first transfer lot of lot 0 is still in process for its operation (0, 0) on M0. Thus, once the first transfer lot completes its operation (0,0) and is transferred to M2, operation (0,1) of the first transfer lot is started immediately without any delay for machine setup. The requirement of one time unit "time out" (transportation time, for example) between operation (0,0) and (0,1) can also be easily observed. Second: operation (3,1) is started on M4 after operation (3,0) of all transfer lots in lot 0 is finished. Once operation (3,1) is completed, the first

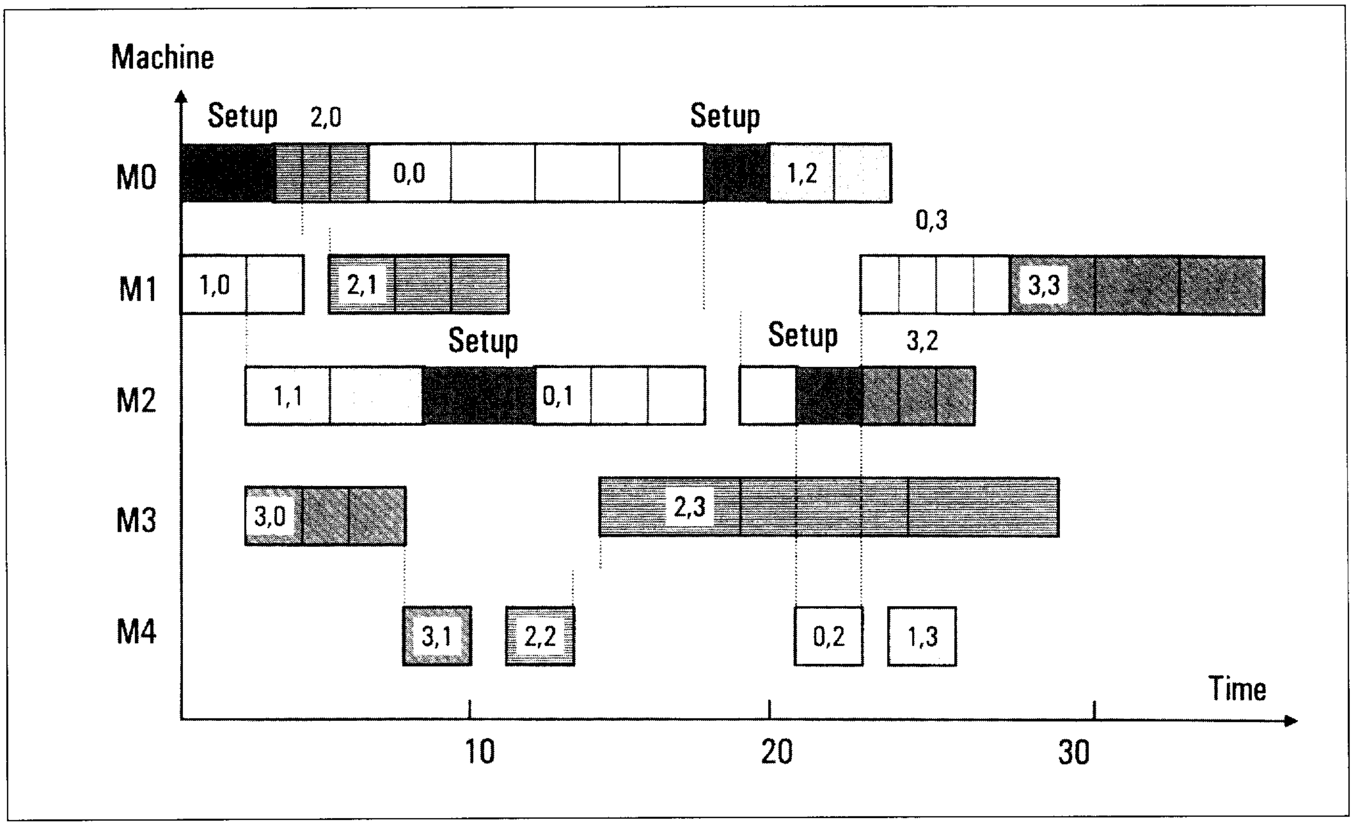

Figure 3
Gantt Chart of Schedule 2: Scheduling Transfer Lots with Various Machine Types

transfer lot of lot 3 is moved to M2 and operation (3,2) is started as soon as M2 is available. These two examples confirm the validity of lot dynamic modeling for different machine categories.

Cases 3 and 4 draw data from a manufacturer producing aircraft/turbine-generator parts. According to the production requirements, the parts to be scheduled are grouped into a number of lots with various due dates and weights. *Table 3* summarizes the test data for Cases 3 and 4.

Case 3 (A Real Problem Using Different Transfer-Lot Sizes)

This case is to demonstrate the capability of the method developed for scheduling real problems using different transfer lot sizes. As given in *Table 3*, the planning horizon in this case is 780 hours, and therefore the number of total multipliers is 17940. First, each part is treated as a transfer lot (transfer lot size = 1). Then the total number of transfer lots is 144, and the average number of transfer lots in a lot is 5.78. The parts include those that have operations requiring setups and can be done on alternative machines; however, these are less than about 5% of the parts. The testing results are shown in *Table 4*. It can be seen that the algorithm does not require long computational time to get high-quality schedules for real problems of this size. To see the impact of transfer lot size on scheduling, every two parts are grouped into a transfer lot (transfer lot size = 2). The total number of transfer lots is 72, and the average number of transfer lots in a lot is 2.88. The testing results are also given in *Table 4*. It can be seen that the larger the transfer lots size, the larger the tardiness (feasible cost), but its solution is closer to the optimum. This implies that smaller transfer lot size usually provides a better solution, as expected, but it requires more computations.

Case 4 (A Larger Real Problem With and Without Time Step Reduction)

A larger problem is tested in this case to further illustrate the capability of the method developed for scheduling large-sized practical problems. The planning horizon is 1170 hours for scheduling a total of 61 lots. First, without using time step reduction (R=1), the number of total multipliers is 29150,

Table 3
Description of Data for Cases 3 and 4: Larger Problems

Case No.	No. of Lots	No. of Parts	No. of Part Types	Avg. No. of Operations/Lot	No. of Machines/ Machine Types	Planning Horizon (hour)
3	25	144	17	8.48	39 / 25	780
4	61	339	17	10.8	44 / 27	1170

Table 4
Results for Case 3: Different Transfer Lot Sizes

Transfer lot size = 1, no. of transfer lots = 144				Transfer lot size = 2, no. of transfer lots = 72			
No. of Iterations	Feasible Cost	Duality Gap	CPU Time (sec.)	No. of Iterations	Feasible Cost	Duality Gap	CPU Time (sec.)
25	86301	21.9%	302	25	98620	17.1%	305
50	86301	18.2%	593	50	96470	12.2%	602

Table 5
Results for Case 4: Time Step Reduction Technique

R = 1 (without time step reduction)					R = 10 (with time step reduction)				
No. of Iterations	Avg. Lead Time	Avg. WIP	Avg. Delay	CPU (sec.)	No. of Iterations	Avg. Lead Time	Avg. WIP	Avg. Delay	CPU (sec.)
6	226	0.28	38	560	6	241	0.28	41	384
24	204	0.27	34.4	2052	24	212	0.27	38.8	1364
36	199	0.266	28.5	3161	36	203	0.272	30.9	2021
48	192	0.255	28	4154	48	203	0.263	29.5	2781

and the results are summarized in *Table 5*. To show the effect of "time step reduction" on scheduling large problems, R=10 is used, decreasing the number of total multipliers to 2915. These results are also summarized in *Table 5*. For comparison, the performance of schedules is measured at several iteration counts: 6, 24, 36, and 48. It is obvious that, at a given iteration number, R=10 needs much less CPU time than R=1 to get a good schedule because of the smaller number of multipliers required by R=10. It can be also seen that R=1 gives a better schedule than R=10 in view of the modeling approximation caused by larger values of R. Measured by practical metrics, these schedules obtained in about 20 minutes for R=10 and in 35 minutes for R=1 look quite reasonable. The duality gap (not included in the table) in this case is significantly larger than previous cases and is still under study. This case implies that for large problems, good schedules can be obtained within a reasonable time by using the time step reduction technique.

Conclusions

The extended lot dynamics model and the backward dynamic programming (BDP) technique have been developed for scheduling with fixed-size transfer lots. The effective handling of transfer lots on machines with setups and machines where all transfer lots in a lot are required to be processed simultaneously is of practical significance. To apply BDP to solve lot subproblems efficiently, the number of multiple completion times associated with each operation beginning time for a lot and the earliest and latest beginning times and completion times for all operations have been be determined. With substates introduced, state transitions determined, and computation-critical DP equations derived for computing stagewise minimum cumulative costs at successor stages, the BDP algorithm is developed to efficiently solve the lot subproblems that contributes to the state-of-the-art scheduling practice. Numerical results indicate that the method can generate high-quality schedules with reasonable computational effort.

Acknowledgment

This work was supported in part by the National Science Foundation under grant DMI 9500037, and the Advanced Technology Center for Precision Manufacturing at the University of Connecticut. The authors would like to thank Dr. Guandong Liu of Ascom Timeplex Corp. and Mr. John Berrigan and Ms. Celia Gonyea of Delta Industries for their invaluable help in this project.

References

1. D. Trietsch and K.R. Baker, "Basic Techniques for Lot Streaming," *Operations Research* (v41, n6, 1993), pp1065-1076.
2. D.H. Kropp and T.L. Smunt, "Optimal and Heuristic Models for Lots Splitting in a Flow Shop," *Decision Sciences* (v21, 1990), pp691-709.
3. B.J. Wagner and G.L. Ragatz, "The Impact of Lot Splitting on Due Date Performance," *Journal of Operations Mgmt.* (v12, 1994), pp13-25.
4. S. Benjaafar, "On Production Batches, Transfer Batches, and Lead Times," *IIE Trans.* (v28, 1996), pp357-362.
5. R.G. Vickson and B.E. Alfredsson, "Two- and Three-Machine Flow Shop Scheduling Problems with Equal Sized Transfer Batches," *Int'l Journal of Production Research* (v30, n7, 1992), pp1551-1574.
6. F.C. Cetinkaya, "Lot Streaming in a Two-Stage Flow Shop with Set-Up, Processing and Removal Times Separated," *Journal of Operation Research Society* (v45, n12, 1994), pp1445-1455.
7. F.R. Jacobs and D.J. Bragg, "The Repetitive Lots: Flow-Time Reduction Through Sequencing and Dynamic Batch Sizing," *Decision Sciences* (v19, 1988), pp281-294.
8. C.A. Glass, J.N.D. Gupta, and C.N. Potts, "Lot Streaming in Three-Stage Production Processes," *European Journal of Operational Research* (v75, 1994), pp378-394.
9. G. Liu and P.B. Luh, "Scheduling Job Shops with Transfer Lots," *Proc.* of 1996 IEEE Int'l Conf. on Robotics and Automation, Minneapolis, April 1996, pp2523-2528.
10. P.B. Luh and D.J. Hoitomt, "Scheduling of Manufacturing Systems Using the Lagrangian Relaxation Technique," *IEEE Trans. on Automatic Control* (v38, n7, 1993), pp1066-1080.
11. 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," *Annals of Operations Research* (v76, 1998), pp233-259.
12. X. Zhao, P.B. Luh, and J. Wang, "The Surrogate Gradient Algorithm for Lagrangian Relaxation Method," *Journal of Optimization Theory and Applications* (v100, n3, 1999), pp699-712.

Appendix A

Derivation of Operation Beginning and Completion Times of Individual Transfer Lots

Assume that there are N_l transfer lots, indexed by $n = 0, 1, ..., N_l-1$, in lot l. Once operation beginning $\{b_{lj}\}$ and completion times $\{c_{lj}\}$ of lot l are determined, these times, $\{b^n_{lj}\}$, $\{c^n_{lj}\}$, of individual transfer lots can be derived as follows.

For all individual transfer lots, if their operation beginning times are given, the corresponding completion times can be determined by the following:

$$c^n_{lj} = b^n_{lj} + t_{ljh} - 1,\ n= 0, 1, ..., N_l-1,\ j=0, 1, ..., J_l-1 \quad (26)$$

First consider that operation (l, j) is processed on standard machines or machines with setups. For the first operation $(l, 0)$, because no intermittent idling times exist in it, the beginning time of an individual transfer lot only relies on the completion time of the predecessor transfer lot, that is,

$$b^n_{l0} = b_{l0} + n\times t_{l0},\ n= 0, 1, ..., N_l-1 \quad (27)$$

For any other operation (l, j), the beginning time of the first transfer lot is the operation beginning time of the lot, and the beginning time of any other transfer lot depends on the completion time of the predecessor transfer lot as well as the completion time of its predecessor operation, that is,

$$b^0_{lj} = b_{lj},\ b^n_{lj} = \max(c^{n-1}_{lj}, c^n_{l,j-1}+ s_{l,j-1}) +1,\ n=1, 2, ..., N_l-1, j=1, 2, ..., J_l-1 \quad (28)$$

Now consider that all transfer lots are required to be processed together simultaneously at (l, j). Since all transfer lots have the same beginning time, beginning time of any individual transfer lot is the operation beginning time of the lot, that is,

$$b^n_{lj} = b_{lj},\ n = 0, 1, ..., N_l-1, j = 0, 1, ..., J_l-1 \quad (29)$$

Appendix B

Determination of Numbers of Multiple Completion Times

For notational consistency with the number of substates introduced in Section 3.2, let $N^c_{lj}+1$ denote the number of possible completion times associated with a given beginning time. For the first operation $(l, 0)$, because all transfer lots are available to be processed, only one completion time corresponds to a given beginning time without any intermittent idling between them, that is, N^c_{l0} equals zero. The parameter t_{lmax} is introduced to represent the maximum processing time among a specified set of operations. First set $t_{lmax} = t_{l0h}$. For any operation (l, j) after $(l, 0)$, if all transfer lots in lot l are required to be processed simultaneously at operation (l, j), then set $N^c_{lj} = 0$ and $t_{lmax} = 0$. For (l, j) processed on standard machines or machines with setups, if $t_{l\max}$ is greater than the processing time t_{ljh} (this implies existing intermittent idling times in (l, j)), then N^c_{lj} can be calculated by the following:

$$N^c_{lj} = (N_l - 1)*(t_{l\max} - t_{ljh}) \quad (30)$$

otherwise, there is no intermittent idling time in (l, j), set $N^c_{lj} = 0$ and $t_{l\max} = t_{ljh}$. Then move to the next operation and repeat the above procedure with the $t_{l\max}$ obtained until the last operation is reached. Using this recursive procedure by proceeding forward, the numbers of possible multiple completion times associated with each beginning time, $\{N^c_{lj}+1\}$, for all operations can be determined. An example is shown in *Figure 4*, and it is clear that the number N^c_{lj} of operation $(l, 0)$, $(l, 2)$, $(l, 3)$, and $(l,4)$ equals zero.

Appendix C

List of Symbols

- a_l Arrival date of required material for lot l
- b_l Desired release time of lot l
- b_l Beginning time of lot l
- b_{lj} Beginning time of operation (l, j)
- b^e_{lj} The earliest start time of operation (l, j)
- b^l_{lj} The latest beginning time of operation (l, j)
- b^n_{lj} Beginning time of operation (l, j) for the nth transfer lot
- b'_{lj} Actual beginning time of operation (l, j) on a machine with setup
- c_l Completion time of lot l
- c_{lj} Completion time of operation (l, j)
- c^e_{lj} The earliest completion time of operation (l, j)
- c^l_{lj} The latest completion time of operation (l, j)
- c^n_{lj} Completion time of operation (l, j) for the nth transfer lot
- d_l Due date of lot l
- E_l Earliness of lot l
- H Set of all machine types
- H_{lj} Eligible machine types for operation (l, j)
- J Objective cost
- J^+ Lagrangian after relaxation of "coupling" constraints
- J_l Total number of operations for lot l
- K Total number of enumeration steps
- k Index of enumeration step, $k = 0, 1, ..., K-1$
- L Total number of lots to be scheduled
- l Lot index, $l = 0, 1, ..., L-1$
- M_{kh} Average number of h type machines available at enumeration step k
- m_{kh} Non-negative slack variable satisfying $0 \le m_{kh} \le M_{kh}$
- N^c_{lj} Number of multiple completion times associated with a beginning time of operation (l, j)
- N_l Number of transfer lots in lot l
- n Index of individual transfer lot, $n = 0, 1, ..., -1$
- R Total number of resolution steps within one enumeration step
- s_{lj} Required "time out" between operation (l, j) and $(l, j+1)$
- T Planning time horizon in resolution increments

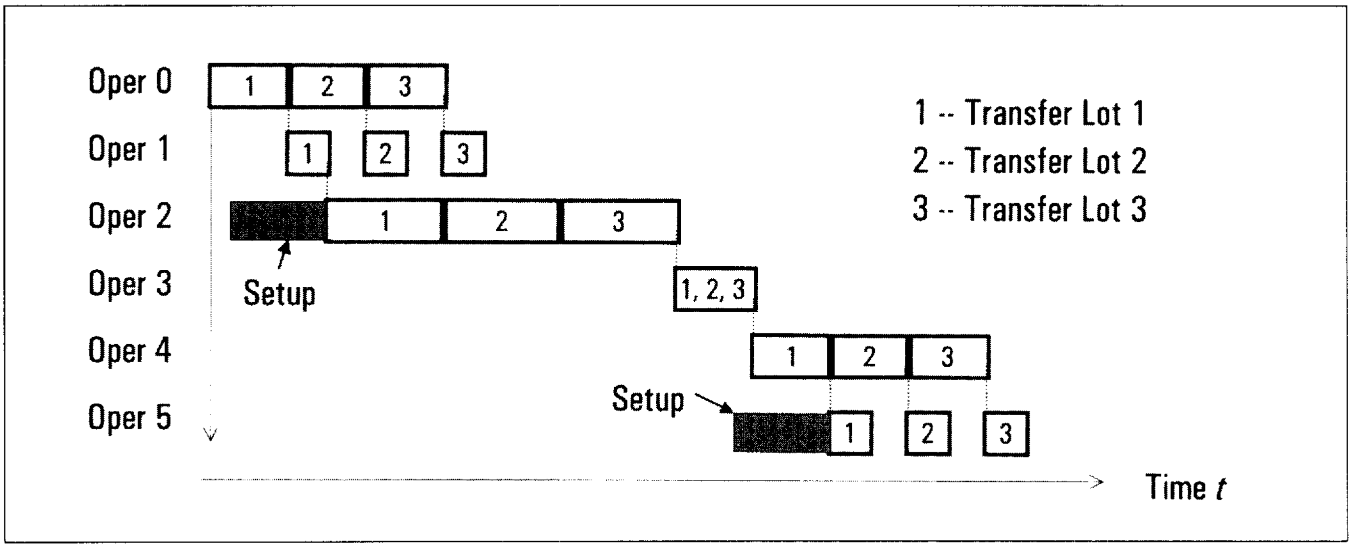

Figure 4
Example for Determining Numbers of Multiple Completion Times

T_l Tardiness of lot l
t Time index in resolution increment, $0 \leq t \leq T-1$
t_{ljh} Processing time of operation (l, j) for each transfer lot of lot l on machine type h
t^s_{ljh} Setup time on machine type h for operation (l, j)
w_l Weight of tardiness penalty of lot l
β_l Weight of earliness penalty of lot l
δ_{ljkh} "Fraction" of the enumeration step k that operation (l, j) is active on machine type h
π_{kh} Lagrange multipliers of machine types h with enumeration step k

Authors' Biographies

Bin Jin is a software engineer with Dialogic Corporation, an Intel Company. He is currently working on computer telephony products, data networking, and telecommunication systems with Windows/Unix programming by using object-oriented technology, C++, C, and Java. He received his MS degree in electrical engineering from University of Connecticut. He has published in the area of control and instrumentation. His research interests include data networking and telecommunication systems and optimization-based manufacturing scheduling.

Peter B. Luh received his BS in electrical engineering from National Taiwan University, his MS in aeronautics and astronautics engineering from MIT, and his PhD in applied mathematics from Harvard University. Since 1980 he has been with the University of Connecticut and currently is director of the Booth Center for Computer Applications and Research and professor in the Department of Electrical and Systems Engineering. He is interested in planning, scheduling, and coordination of design and manufacturing activities and has developed a near-optimal and efficient schedule generation and reconfiguration methodology to improve on-time delivery of products and reduce work-in-progress inventory. He is also interested in schedule and transaction optimization for power systems and has developed near-optimal and efficient unit commitment, hydrothermal coordination, and power transaction methodologies to minimize total fuel costs. He owns one US patent and three Best Paper Awards. Dr. Luh is a fellow of IEEE and a member of the Connecticut Academy of Science and Engineering. He has been on the editorial board of IEEE Transactions on Robotics and Automation since 1990, is an editor of IEEE Robotics and Automation Magazine, and was an associate editor for IEEE Transactions on Automatic Control.

Lakshman S. Thakur received his BSc from Bombay University in mathematics and physics, his doctorate from Columbia University in operations research and industrial engineering, and is an associate professor of operations and information management at the University of Connecticut. He has more than 20 years of experience in teaching, consulting, and research on optimization under resource constraints. He has been consultant to IBM on manpower planning with risk assessment and product warranty systems, as well as a senior consultant and director of management science in a consulting organization. He is an associate editor of Naval Research Logistics. Dr. Thakur has been a visiting professor of operations research at the Yale University School of Management, and he has published in Management Science, Mathematics of Operations Research, SIAM Journal on Applied Mathematics, SIAM Journal on Optimization and Control, Journal of Mathematical Analysis and Applications, Naval Research Logistics, and other journals. His primary research interests are in the development and applications of linear, nonlinear, and integer optimization methods in management science and spline function approximation in mathematics. His current research focuses on production scheduling, product design, and facility location problems. His research on scheduling (with Dr. Peter B. Luh) is supported by National Science Foundation grants. He is a full member of INFORMS and the Mathematical Programming Society.