IIE Transactions (2003) **35**, 973–985
Copyright © "IIE"
ISSN: 0740-817X print
DOI: 10.1080/07408170390227550

Optimization-based manufacturing scheduling with multiple resources, setup requirements, and transfer lots

DONG CHEN[1], PETER B. LUH[2], LAKSHMAN S. THAKUR[3] and JACK MORENO JR.[4]

[1] *Aristeia Capital L.L.C., 381 Fifth Avenue, 6th Floor, New York, NY 10016, USA*
E-mail: dong_chen100@hotmail.com
[2] *Department of Electrical and Computer Engineering, and* [3] *School of Business Administration, University of Connecticut, Storrs, CT 06269, USA*
E-mail: Peter.Luh@uconn.edu or thakur@uconn.edu
[4] *J. M. Products Inc., Southbury, CT 06488, USA*

Received April 1999 and accepted March 2003

The increasing demand for on-time delivery of products and low production cost is forcing manufacturers to seek effective schedules to coordinate machines and operators so as to reduce costs associated with labor, setup, inventory, and unhappy customers. This paper presents the modeling and resolution of a job shop scheduling system for J. M. Products Inc., whose manufacturing is characterized by the need to simultaneously consider machines and operators, machines requiring significant setup times, operators of different capabilities, and lots dividable into transfer lots. These characteristics are typical for many manufacturers, difficult to handle, and have not been adequately addressed in the literature. In our study, an integer optimization formulation with a separable structure is developed where both machines and operators are modeled as resources with finite capacities. Setups are explicitly considered following our previous work with additional penalties on excessive setups. By analyzing transfer lot dynamics, transfer lots are modeled by using linear inequalities. The objective is to maximize on-time delivery of products, reduce inventory, and reduce the number of setups. By relaxing resource capacity constraints and portions of precedence constraints, the problem is decomposed into smaller subproblems that are effectively solved by using a novel dynamic programming procedure. The multipliers are updated using the recently developed surrogate subgradient method. A heuristic is then used to obtain a feasible schedule based on subproblem solutions. Numerical testing shows that the method generates high quality schedules in a timely fashion.

1. Introduction

The increasing demand for on-time delivery of products and low production cost is forcing manufacturers to seek effective schedules to coordinate machines and operators so as to reduce costs associated with labor, setup, inventory, and unhappy customers. This paper presents the modeling and resolution of a job shop scheduling system for J. M. Products Inc., a typical small manufacturer of mechanical components with about 10 people in Connecticut. Parts are processed in lots, and all parts within a lot must be completed on a machine before that machine can process another lot. A lot, however, may be divided into multiple "transfer lots," each of which can move to the next processing stage as soon as all parts within that transfer lot are finished. This lot splitting allows individual transfer lots to be processed concurrently at consecutive operation stages, and can significantly reduce manufacturing lead times and lower inventory levels. Some machines need to be set up before they can process specific lots, and setups are "group-dependent" in the sense that a different setup is needed when processing is switched from a lot of a particular "group" of part types to a lot of a different "group" of part types. Excessive setups are highly undesirable since they lead to undesirable setup costs and increase the chance of rework and scrap. In addition, in view of limited personnel, operators of different capabilities (e.g., operators for machine setups or for processing) need to be efficiently scheduled and well coordinated with machines. The objective is to deliver lots on-time, to reduce the Work-In-Process (WIP) inventory, and to decrease the number of setups. These characteristics are typical for many manufacturers, difficult to handle, and have not been adequately addressed in the literature.

2. Literature review

2.1. *Multiple resources*

Several studies on multiple resource scheduling where an operation may require multiple resources (e.g., a machine and an operator) have been reported in the literature

(Treleven and Elvers, 1985; Luh, Liu and Moser, 1999). Most of them, however, were based on heuristics (Gargeya and Deane, 1996). These methods have the merit of being computationally efficient and can be applied to problems of practical sizes. The results obtained, however, are often of questionable quality, and it is very difficult to systematically improve the results. Optimization-based methods were presented only in a few studies. For examples, multiple resource scheduling to minimize weighted flow times was discussed in Dobson and Karmarkar (1993). Since a "disjunctive formulation" with a non-separable structure was used, a large problem cannot be decomposed into smaller subproblems to efficiently obtain solutions. Scheduling a job shop with multiple resources was presented in Chen and Hsia (1994). Without considering setups, operators were essentially modeled as machines and the problem was solved by using Lagrangian relaxation following our previous work (Czerwinski and Luh, 1994).

2.2. *Group-dependent setups*

In an earlier survey on machine setups, 70% of industrial schedulers reported that they had to deal with "sequence-dependent setups" in which the setup time for a lot depends on what is processed before that lot (Panwalkar *et al.*, 1973). Scheduling with sequence-dependent setups is recognized as being very difficult, and most existing results in the literature focus on either a single machine or several identical machines (Kim and Bobrowski, 1994; Ovacik and Uzsoy, 1994; Young *et al.*, 1997). Only a few studies addressed sequence-dependent flow shops or job shops, and branch-and-bound and heuristics were the predominant methods. The computation time of a branch-and-bound method, however, increases drastically as the problem size increases (Brucker and Thiele, 1996). "Group-dependent setups" are a special kind of sequence-dependent setups where a setup is needed when processing is switched from one "group" of parts to another (groups could be defined based on selected fields of the parts' Group Technology code), and the setup time depends only on the group which is being set up. A job shop scheduling method considering group-dependent setups based on Lagrangian relaxation was recently presented in Luh *et al.* (1998) where operators were not considered. To reduce the number of setups, most of the currently existing methods are based on either heuristics (Soumen and Cheryl, 1993) or a branch-and-bound approach (e.g., Liao and Chuang (1996) for a single facility).

2.3. *Transfer lots*

The handling of lot splitting can be generally classified into two categories: (i) treating individual transfer lots as independent scheduling units; or (ii) by treating each lot as an independent unit. The first category of methods that treats each transfer lot as an independent unit has been frequently reported in the literature (Vickson and Alfredsson, 1992; Trietsch and Baker, 1993). These methods require a large number of decision variables (for individual transfer lots), and a special set of constraints to ensure that all transfer lots within a lot are completed before a machine can process another lot. These lead to significant mathematical difficulties except for very small problems (Dobson and Karmarkar, 1989). The second category of methods that treats each lot as an independent scheduling unit has fewer decision variables and constraints than the first one. An optimization-based formulation that elegantly describes lot dynamics was recently developed in Liu and Luh (1996). The model, however, includes nonlinear equalities making the solution methodology difficult to implement.

2.4. *Scope of this paper*

In this paper, both machines and operators are modeled as resources with finite capacities. Operators may have different capabilities, and machine setups are explicitly modeled following our previous work (Luh *et al.*, 1998) with additional penalties on excessive setups. By treating each lot as an independent scheduling unit and analyzing transfer lot dynamics, transfer lots are modeled using linear inequalities instead of the nonlinear equalities as in Liu and Luh (1996). A separable formulation considering all these features is presented in Section 3. By relaxing resource capacity constraints and portions of precedence constraints, this problem is decomposed into "lot subproblems" and "group subproblems." The lot subproblem is solved by using a novel Dynamic Programming (DP) procedure which is much simpler than that presented in Liu and Luh (1996). The group subproblem is solved by using the standard DP with penalties on excessive setups embedded in state transition costs without requiring much additional computation. The Lagrangian multipliers are updated by using the recently developed Surrogate Subgradient Method (Zhao *et al.*, 1999). A heuristic procedure is developed to adjust subproblem solutions to obtain a feasible schedule satisfying all constraints as presented in Section 4. The method has been implemented by using the object-oriented programming language C++ with a Microsoft Access user interface, and numerical testing shows that it generates high quality schedules in a timely fashion. Through simultaneous consideration of machines and operators, machines and operators are well coordinated to facilitate the smooth flow of lots through the system. The explicit modeling of setups and the associated penalties encourage lots with the same setup requirements to be processed back-to-back to avoid excessive setups. The linear description of transfer lots also significantly reduces the implementation complexity without sacrificing solution quality.

3. Problem formulation

The following formulation is built on our previous work on job shop scheduling with group-dependent setups (Luh *et al.*, 1998) and with transfer lots (Liu and Luh, 1996). It has the following new features: (i) non-symmetric modeling of machines and operators, while an operator can supervise several operations at the same time; (ii) additional penalties on excessive setups in the objective function; and (iii) linear modeling of transfer lots. For clarity, the preliminaries are presented first.

3.1. *Notation and general description*

3.1.1. *Machines and operators*

In the formulation, K discrete time units are considered with index k ranging from zero to $K-1$. There are H machine types, and the number of type h machines $(1 \leq h \leq H)$ at time k is given and denoted as M_{kh}. Based on their skills, operators are classified into O operator types, and the number of type o operators $(1 \leq o \leq O)$ at time k is given and denoted as O_{ko}. These operators can roughly be divided into operation operators who process specific operations but cannot set up machines, and setters who set up certain machines and process specific operations.

3.1.2. *Lots and transfer lots*

There are L production lots to process, indexed by l $(1 \leq l \leq L)$. Lot l consists of a number of parts of the same part type, and has its arrival time a_l, due date d_l, and priority (or weight) w_l. Lot l can be divided into multiple equal-sized transfer lots, and the number of transfer lots is given and denoted as N_l. Lot l requires a sequence of J_l operations for completion, and operation j $(1 \leq j \leq J_l)$ of lot l is denoted as (l, j).

Operation (l, j) has to be processed on a machine of type h belonging to a given set of "eligible" machine types H_{lj}. The processing time t_{ljh} of (l, j) for a transfer lot on a type h machine is assumed given, and the processing may require a certain percent of attention m_{ljo} of a type o operator belonging to a given set of "eligible" operator types O_{lj}. The first operation of lot l can be started only after the arrival of the order or appropriate raw materials. Similarly, operation (l, j) of a transfer lot can be started on a machine after the transfer lot has arrived from the predecessor operation $(l, j-1)$, and this machine has finished the preceding transfer lot. If the predecessor operation $(l, j-1)$ requires a longer processing time, an intermittent idling may occur as illustrated in Fig. 1 for a lot with three transfer lots.

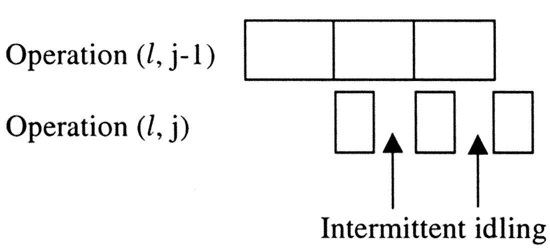

Fig. 1. Intermittent idling between transfer lots.

Consequently, the derivation of operation completion time is quite complicated as will be detailed later.

3.1.3. *Group-dependent setups*

As mentioned earlier, some machines need to be set up before they can process specific lots. All the lots with the same setup requirements for a particular operation are classified as a "group." Operations of group g on type h machines are processed in several "runs," where operations within a run share a single setup. The nth run is denoted as (h, g, n), and the setup for run (h, g, n) requires 100% attention of a type o setter belonging to a given set of "eligible" operator types O_{hg} for the S_{hg} amount of time.

3.2. *Modeling of resource capacity constraints*

With the concept of runs, machine capacity constraints with group-dependent setups can be described as follows. Operations without setup requirements and runs occupy machines, while runs provide "virtual" facilities to host operations with setup requirements.

3.2.1. *Machine capacity constraints*

The number of active operations without setup requirements and the number of active runs cannot exceed machine capacity, i.e.,

$$\sum_{(l,j)\in\bar{S}} \delta_{ljkh} + \sum_{g,n} \phi_{khgn} + M_{kh}, \forall\, k, h. \tag{1}$$

In the above, $\bar{S}$ denotes the set of operations that do not need a setup. The operation variable δ_{ljkh} equals one if operation (l, j) is assigned to a type h machine at time k, and zero otherwise, i.e.,

$$\delta_{ljkh} = \begin{cases} 1, & \text{if operation } (l, j) \text{ is assigned to machine} \\ & \quad \text{type } h \text{ and } b_{lj} \leq k \leq c_{lj}, \\ 0, & \text{otherwise,} \end{cases} \tag{1a}$$

where b_{lj} and c_{lj} are the beginning time and completion time of operation j of lot l, respectively. The run variable φ_{khgn} equals one if run (h, g, n) is active at time k, and zero otherwise, i.e.,

$$\varphi_{khgn} = \begin{cases} 1, & \text{if run } (h, g, n) \text{ is active at time } k, \\ & \quad \text{i.e., } b_{hgn} \leq k \leq c_{hgn}, \\ 0, & \text{otherwise,} \end{cases} \tag{1b}$$

where b_{hgn} and c_{hgn} are the beginning time and completion time of run (h, g, n), respectively.

3.2.2. *Group constraints*

Operations with setup requirements cannot be processed unless one of the related runs is active and has been set up, i.e.,

$$\sum_{(l,j)\in S} \delta_{ljkh} \leq \sum_{n} \phi_{khgn} - \sum_{n} \zeta_{khgn}, \forall\, k,\ h,\ g, \tag{2}$$

where S denotes the set of operations that require setups, and setup variable ζ_{khgn} equals one if run (h, g, n) is being set up at time k, and zero otherwise, i.e.,

$$\zeta_{khgn} = \begin{cases} 1, & \text{if run } (h, g, n) \text{ is being set up at time } k, \\ & \quad \text{i.e., } b_{hgn} \le k \le c_{hgn}, \\ 0, & \text{otherwise.} \end{cases} \tag{2a}$$

Since setup and processing within a run may be performed by different operators, the concept of a group is not directly applicable to operators. This makes the modeling of operators and machines unsymmetrical when setups are involved.

3.2.3. *Operator capacity constraints*

For each operator type, the sum of an operator's attention for operations and setups cannot exceed operator capacity, i.e.,

$$\sum_{lj} m_{ljo}\delta_{ljko} + \sum_{gn} \zeta_{kogn} \le O_{ko}, \forall\, k, o. \tag{3}$$

In the above, operation variable δ_{ljko} equals one if operation (l, j) is performed by a type o operator at time k, and zero otherwise, i.e.,

$$\delta_{ljko} = \begin{cases} 1, & \text{if operation } (l, j) \text{ is performed by operator} \\ & \quad o \text{ and } b_{lj} \le k \le c_{lj}\ , \\ 0, & \text{otherwise.} \end{cases} \tag{3a}$$

Setup variable ζ_{kogn} equals one if run n of group g is being set up by a type o operator at time k, and zero otherwise, i.e.,

$$\zeta_{kogn} = \begin{cases} 1, & \text{if run } n \text{ of group } g \text{ is being setup by a type } o \\ & \quad \text{operator at time } k, \text{ i.e., } b_{ogn} \le k \le c_{ogn}, \\ 0, & \text{otherwise.} \end{cases} \tag{3b}$$

Since $0 \le m_{ljo} \le 100\%$, an operator's attention may be shared by several operations at the same time.

3.3. *Precedence constraints and completion time constraints*

In view of the existence of transfer lots and the concomitant intermittent idling times between transfer lots, precedence relationships for the problem are complicated. They will be presented under the categories of operation precedence constraints, arrival time constraints, completion time constraints, and run sequence constraints below.

3.3.1. *Operation precedence constraints*

The first transfer lot cannot be started before its predecessor operation has been completed plus any required "time-out," i.e.,

$$b_{l,j-1} + t_{l,j-1,h} + s_{l,j-1} \le b_{lj},\ \forall l, j > 1, h \in H_{ij}, \tag{4}$$

where $s_{l,j-1}$ is any required "time-out" between operation $j-1$ and its succeeding operation j of lot l.

3.3.2. *Arrival time constraints*

The processing of a lot cannot be started before the arrival of the order or appropriate raw materials, i.e.,

$$a_l \le b_{l1},\ \forall l. \tag{5}$$

3.3.3. *Completion time constraints*

Each lot must be assigned a sufficient amount of time to process all its transfer lots on a machine belonging to an eligible machine type, i.e.,

$$b_{lj} + N_l \cdot t_{ljh} - 1 \le c_{lj},\ \forall (l, j), h \in H_{ij}. \tag{6}$$

In view of possible intermittent idling times, the lot completion time is not a simple sum of the lot beginning time and the required processing time. Nevertheless, it is clear that the last transfer lot can be processed only after its predecessor operation has been completed plus any required "time-out," i.e.,

$$c_{l,j-1} + t_{ljh} + s_{l,j-1} \le c_{lj},\ \forall (l, j), h \in H_{ij}, \tag{7}$$

with $c_{l0} \equiv -1$, $s_{l0} \equiv 0$.

The above two linear inequality constraints succinctly describe the dynamics of transfer lots, and are key to our later derivation. Note that in Liu and Luh (1996), the completion times were modeled as

$$c_{lj} = \max\{b_{lj} + N_l \cdot t_{ljh} - 1,\ c_{l,j-1} + s_{l,j-1} + t_{ljh}\},$$
$$\forall (l, j), h \in H_{ij}, \tag{8}$$

where it is implicitly assumed that the last transfer lot of a lot must be processed as early as possible. This assumption is a special case of our linear modeling Equations (6) and (7). Furthermore, since Equation (8) is nonlinear, the solution methodology presented in Liu and Luh (1996) was very complicated.

3.3.4. *Run sequencing constraints*

The runs of a particular group are assumed to be processed in the ascending order of their run numbers, i.e.,

$$c_{hgn} + 1 \le b_{hg,n+1},\ \forall (h, g, n). \tag{9}$$

3.4. *Objective function*

The objectives to be achieved are on-time delivery of lots, a low WIP inventory, and a small number of setups. Since direct minimization of the number of setups without introducing additional variables is difficult, our idea to achieve a small number of setups is to first assume that a sufficient number of runs is given and then penalize undesirable runs. With a penalty on the duration of a run (h, g, n), some runs may only include a setup, i.e., $c_{hgn} = b_{hgn} + S_{hg} - 1$, and perform host operations, implying that the run in fact does not exist. The objective is thus translated to the minimization of penalties on lot tardiness, on releasing raw materials too early, and on durations of undesirable runs, i.e.,

$$\min_{\{b_{lj}, c_{lj}, o_{lj}, h_{lj}, b_{hgn}, c_{hgn}, o_{hg}\}} J,$$

with

$$J \equiv \sum_l \left(w_l T_l^2 + \beta_l E_l^2\right) + \sum_{hgn} \alpha_{hgn} U(c_{hgn} - b_{hgn} - S_{hg}), \tag{10}$$

subject to the constraints of Equations (1)–(7) and (9).

In the above, tardiness T_l for lot l is the amount of overdue time, i.e., max(0, $c_l - d_l$). For a given lot due date d_l, a desired lot start time b_{ld} can be roughly estimated, and earliness E_l is then defined as the amount that lot beginning time leads the desired start time, i.e., max(0, $b_{ld} - b_l$) (see, e.g., Czerwinski and Luh (1994)). Weights w_l and β_l reflect the importance of meeting on-time completion and low WIP inventory, respectively. The unit step function $U(x)$ equals one if $x \geq 0$ and zero otherwise, and $U(c_{hgn} - b_{hgn} - S_{hg})$ thus characterizes whether or not the run (h, g, n) exists. Weight α_{hgn} is a penalty coefficient, and is set to zero or a small value for acceptable runs and to a large value for undesirable runs. This indirect penalization on a large number of setups keeps the objective additive without introducing additional variables.

The overall problem is to minimize Equation (10) subject to Equations (1)–(7) and (9). The key decision variables are the operation beginning times $\{b_{lj}\}$, the operation completion times $\{c_{lj}\}$, the machine type to be used $\{h \in H_{lj}\}$, the operator type to be assigned $\{o \in O_{lj}\}$, the run beginning time $\{b_{hgn}\}$, the run completion time $\{c_{hgn}\}$, and the setup operator type to be assigned $\{o \in O_{hg}\}$. Once these variables are determined, other variables can be easily derived. Since Equations (4)–(7) and (9) are linear, and Equations (1)–(3) and (10) are additive in terms of decision variables, the formulation is "separable." Lagrangian relaxation can thus be effectively applied as presented in the following section.

4. Solution methodology

Similar to the pricing concept of a market economy, the Lagrangian Relaxation (LR) method replaces "hard" coupling constraints (e.g., resource capacity constraints) by the payment of certain "prices" (i.e., Lagrange multipliers) for the use of a machine and/or an operator for each time unit. The original problem can thus be decomposed into many smaller subproblems. These subproblems are much easier to solve as compared to the original problem, and their solutions can be efficiently obtained by using DP. After these subproblems are solved, the multipliers are iteratively adjusted based on the degrees-of-constraint violation following again the market economy mechanism. Subproblems are then resolved based on the new set of multipliers. In mathematical terms, the "dual function" is maximized in this multiplier updating process, and the values of the dual function are lower bounds to the optimal feasible cost. Since the coupling constraints have been relaxed, the solutions of individual subproblems may not constitute a feasible schedule. Therefore, at the termination of this multiplier updating process, a simple heuristic is used to adjust subproblem solutions to provide a feasible schedule satisfying all constraints. The quality of the feasible schedule can be quantitatively evaluated by comparing its cost to the largest lower bound provided by the dual function.

In Liu and Luh (1996), resource capacity constraints are relaxed, and the resulting subproblems are subject to precedence constraints with completion times described by the nonlinear relationship of Equation (8). This nonlinear relationship makes the DP procedure for subproblems very complicated. As a result, it is difficult to implement the algorithm and to maintain the software. In the following, resource capacity constraints and the precedence constraints related to operation completion times, Equation (7), are relaxed by using Lagrange multipliers, and the relaxed problem is decomposed into individual lot and group subproblems. The group subproblem is solved by using the standard DP procedure with penalties on excessive setups embedded in the state transition costs without requiring much additional computation. By analyzing the lot dynamics, a novel DP procedure is developed to solve lot subproblems. As compared to Liu and Luh (1996), this DP procedure is much simpler, and can be easily implemented based on our previous modules for job shop scheduling (Luh *et al.*, 1998). This DP procedure also requires much less CPU time for each iteration, although more iterations may be needed to achieve the same schedule quality since more constraints have been relaxed.

4.1. *Lagrangian function*

Since the resource capacity constraints, Equations (1)–(3) and the precedence constraints related to completion time, Equation (7), are "hard" coupling constraints, they are relaxed by using non-negative Lagrange multipliers $\{\pi_{ko}\}$, $\{\pi_{kh}\}$, $\{\gamma_{gkh}\}$, and $\{\lambda_{lj}\}$, respectively, and the Lagrangian L is formed as:

$$\begin{aligned} L \equiv & \sum_l \left(w_l T_l^2 + \beta_l E_l^2\right) + \sum_{hgn} \alpha_{hgn} U(c_{hgn} - b_{hgn} - S_{hg}) \\ & + \sum_{k,h \in H_{lj}} \left\{ \pi_{kh} \left[\sum_{l,j \in S} \delta_{ljkh} + \sum_{g,n} \varphi_{khgn} - M_{kh} \right] \right\} \\ & + \sum_{g,k,h \in H_{lj}} \left\{ \gamma_{gkh} \left[\sum_{l,j \in S} \delta_{ljkh} + \sum_n \zeta_{khgn} - \sum_n \varphi_{khgn} \right] \right\} \\ & + \sum_{k,o \in O_{lj}} \left\{ \pi_{ko} \left[\sum_{l,j} m_{ljo} \delta_{ljko} + \sum_{gn} \zeta_{kogn} - O_{ko} \right] \right\} \\ & + \sum_{l,j} \lambda_{lj} (c_{l,j-1} + t_{ljh} + s_{lj} - c_{lj}). \end{aligned} \tag{11}$$

After regrouping relevant terms within L, the relaxed problem is decomposed into lot and group subproblems which can be solved separately as discussed in the following section.

4.2. *Lot subproblems*

Collecting all the terms in Equation (11) related to lot l and using Equations (1a), (1b), (2a), (3a), and (3b) leads to the following lot subproblems:

$$\min_{\{b_{lj}, c_{lj}, o_{lj}, h_{lj}\}} L_l,$$

with

$$L_l \equiv w_l T_l^2 + \beta_l E_l^2 + \sum_{j=0}^{J_l-1} L_{lj}(b_{lj}, c_{lj}, o_{lj}, h_{lj}), \tag{12}$$

where if (l, j) does not need a setup then:

$$L_{lj}(b_{lj}, c_{lj}, o_{lj}, h_{lj}) \equiv \sum_{k=b_{lj}}^{c_{lj}} m_{ijo}\pi_{ko} + \sum_{k=b_{lj}}^{c_{lj}} \pi_{kh} + \lambda_{lj} t_{ljh} + (\lambda_{l,j+1} - \lambda_{lj})c_{lj} - \lambda_{l1}\Delta_{lj}, \tag{13}$$

and if (l, j) requires a setup then

$$L_{lj}(b_{lj}, c_{lj}, o_{lj}, h_{lj}) \equiv \sum_{k=b_{lj}}^{c_{lj}} m_{ijo}\pi_{ko} + \sum_{k=b_{lj}}^{c_{lj}} \gamma_{gkh} + \lambda_{lj} t_{ljh} + (\lambda_{l,j+1} - \lambda_{lj})c_{lj} - \lambda_{l1}\Delta_{lj}, \tag{14}$$

subject to the arrival time constraint, Equation (5), operation precedence constraints, Equation (4), and completion time constraints, Equation (6). In the above, $\lambda_{lJ_{l+1}} = 0$, and Δ_{lj} equals one if (l, j) is first operation of lot l and zero otherwise. In Equations (13) and (14), $\sum_{k=b_{lj}}^{c_{lj}} m_{ijo}\pi_{ko}$ reflects the operator utilization cost, $\sum_{k=b_{lj}}^{c_{lj}} \pi_{kh}$ the machine utilization cost, $\sum_{k=b_{lj}}^{c_{lj}} \gamma_{gkh}$ the cost for occupying a run, and $\lambda_{lj} t_{ljh} + (\lambda_{l,j+1} - \lambda_{lj})c_{lj} - \lambda_{lj}\Delta_{lj}$ the cost for violating Equation (7). A lot subproblem thus reflects the balance between tardiness and earliness penalties, operator utilization costs, machine utilization costs, the costs for occupying runs, and costs for violating the precedence constraints related to completion times.

4.3. *DP for solving lot subproblems*

DP has been used to solve part subproblems for standard job shop scheduling, e.g., Luh *et al.* (1998). For the lot subproblem under consideration, in view that both operation beginning and completion times are decision variables, the following equalities are used to separate the costs associated with operation beginning and completion times so as to solve them individually:

$$L_{lj}(b_{lj}, c_{lj}, o_{lj}, h_{lj}) = L_{lj}^{\mathrm{b}}(b_{lj}, o_{lj}, h_{lj}) + L_{lj}^{\mathrm{c}}(c_{lj}, o_{lj}, h_{lj}), \tag{15}$$

where if (l, j) does not need a setup; then

$$L_{lj}^{\mathrm{b}}(b_{lj}, o_{lj}, h_{lj}) \equiv -\sum_{k=0}^{b_{lj}-1} m_{ijo}\pi_{ko} - \sum_{k=0}^{b_{lj}-1} \pi_{kh} + \lambda_{lj} t_{ljh}, \tag{16}$$

and if (l, j) requires a setup; then

$$L_{lj}^{\mathrm{b}}(b_{lj}, o_{lj}, h_{lj}) \equiv -\sum_{k=0}^{b_{lj}-1} m_{ijo}\pi_{ko} - \sum_{k=0}^{b_{lj}-1} \gamma_{gkh} + \lambda_{lj} t_{ljh}, \tag{17}$$

and if (l, j) does not need a setup; then

$$L_{lj}^{\mathrm{c}}(c_{lj}, o_{lj}, h_{lj}) \equiv \sum_{k=0}^{c_{lj}} m_{ijo}\pi_{ko} + \sum_{k=0}^{c_{lj}} \pi_{kh} + (\lambda_{l,j+1} - \lambda_{lj})c_{lj} - \lambda_{l1}\Delta_{lj}, \tag{18}$$

and if (l, j) requires a setup; then

$$L_{lj}^{\mathrm{c}}(c_{lj}, o_{lj}, h_{lj}) \equiv \sum_{k=0}^{c_{lj}} m_{ijo}\pi_{ko} + \sum_{k=0}^{c_{lj}} \gamma_{gkh} + (\lambda_{l,j+1} - \lambda_{lj})c_{lj} - \lambda_{l1}\Delta_{lj}, \tag{19}$$

In the DP procedure, each operation has two stages, one corresponding to operation beginning times where the states are possible beginning times, and the other operation completion time where the states are possible completion times. To clearly describe the DP procedure, the schematic for a lot with three operations is shown in Fig. 2.

The DP procedure starts with operation 3, the last operation of the lot. The stagewise cost for the completion stage can be calculated by using Equations (18) and (19), and for the beginning stage by Equations (16) and (17). The cost for the entire operation for a particular beginning time is the sum of the beginning stagewise cost and the minimum of the completion stagewise cost subject to the completion time constraint, Equation (6). Generally, the procedure is as follows:

$$\begin{aligned} V_{lJ_l}&(b_{lJ_l}, o_{lJ_l}, h_{lJ_l}) \\ &\equiv \min_{c_{lJ_l}} \left\{ L_{lJ_l}^{\mathrm{b}}(b_{lJ_l}, o_{lJ_l}, h_{lJ_l}) + L_{lJ_l}^{\mathrm{c}}(c_{lJ_l}, o_{lJ_l}, h_{lJ_l}) + w_l T_l^2 \right\}, \\ &\equiv L_{lJ_l}^{\mathrm{b}}(b_{lJ_l}, o_{lJ_l}, h_{lJ_l}) + \min_{c_{lJ_l}} \left\{ L_{lJ_l}^{\mathrm{c}}(c_{lJ_l}, o_{lJ_l}, h_{lJ_l}) + w_l T_l^2 \right\}, \end{aligned} \tag{20}$$

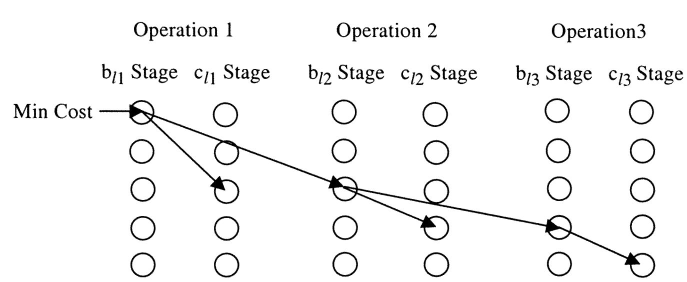

Fig. 2. DP procedure for a lot subproblem.

subject to Equation (6). The DP procedure then moves to operation 2. Similar to operation 3, the cost for operation 2 for a particular beginning time is the sum of the beginning stagewise cost and the minimum of the completion stagewise cost subject to the completion time constraints, Equation (6). The cumulative cost for operations 2 and 3 is then the sum of the operation 2 cost and the minimum of the operation 3 cost among possible operation 3 beginning times and eligible machine and operator types subject to the operation precedence constraints, Equation (4). The DP procedure then moves to operation 1, and the process repeats. Generally, the DP procedure is as follows:

$$\begin{aligned} V_{lj}(b_{lj}, o_{lj}, h_{lj}) &\equiv \beta_l E_l^2 \Delta_{lj} + L_{lj}^{\mathrm{b}}(b_{lj}, o_{lj}, h_{lj}) + \min_{c_{lj}} L_{lj}^{\mathrm{c}}(c_{lj}, o_{lj}, h_{lj}) \\ &\quad + \min_{\{b_{l,j+1}, o_{l,j+1}, h_{l,j+1}\}} V_{l,j+1}(b_{l,j+1}, o_{l,j+1}, h_{l,j+1}), \\ &\qquad 1 \le j \le J_l - 1, \end{aligned} \tag{21}$$

subject to Equations (6) and (4). The optimal L_l^* is then obtained as the minimal cumulative cost for the first operation, subject to the arrival time constraint, Equation (5). Finally, the optimal beginning times, completion times, and the corresponding machine and operator types can be obtained by forward tracing the stages. This DP procedure is unconventional because of the complex relationships among beginning and completion times of adjacent stages as described by Equations (4), (6) and (7). The complexity is $O(2K \sum_j |H_{lj}| \times |O_{lj}|)$, where $|x|$ is the cardinality of set x.

4.4. *Group subproblems and their solutions*

Collecting all the terms in Equation (11) related to group g on machine type h leads to:

$$\min_{\{b_{hgn}, c_{hgn}, o_{hgn}\}} L_{hg},$$

with

$$\begin{aligned} L_{hg} &\equiv \sum_n \sum_{k=b_{hgn}}^{b_{hgn}+S_{hg}-1} \pi_{ko} + \sum_n \sum_{k=b_{hgn}}^{c_{hgn}} \pi_{kh} - \sum_n \sum_{k=b_{hgn}+S_{hg}}^{c_{hgn}} \gamma_{khg} \\ &\quad + \sum_n \alpha_{hgn} U(c_{hgn} - b_{hgn} - S_{hg}), \end{aligned} \tag{22}$$

subject to the run sequencing constraints, Equation (9). In Equation (22),

$$\sum_{k=b_{hgn}}^{b_{hgn}+S_{hg}-1} \pi_{ko},$$

reflects the setter utilization cost,

$$\sum_{k=b_{hgn}}^{c_{hgn}} \pi_{kh},$$

the cost for occupying a machine, and

$$-\sum_{k=b_{hgn}+S_{hg}}^{c_{hgn}} \gamma_{khg},$$

the value for hosting operations. A group subproblem thus reflects the balance among setter utilization costs, costs for occupying a machine, values for hosting operations, and penalties on excessive setups.

Since the processing time for a run is unspecified, the beginning and completion times are all decision variables. The following equality is employed to separate the cost associated with a run into three portions related to run beginning time, run completion time, and the penalty for an undesirable run:

$$\begin{aligned} &\sum_{k=b_{hgn}}^{b_{hgn}+S_{hg}-1} \pi_{ko} + \sum_{k=b_{hgn}}^{c_{hgn}} \pi_{kh} - \sum_{k=b_{hgn}+S_{hg}}^{c_{hgn}} \gamma_{gkh} \\ &\quad + \alpha_{hgn} U(c_{hgn} - b_{hgn} - S_{hg}) \\ &= \left\{ \sum_{k=b_{hgn}}^{b_{hgn}+S_{hg}-1} \pi_{ko} - \sum_{k=0}^{b_{hgn}-1} \pi_{kh} + \sum_{k=0}^{b_{hgn}+S_{hg}-1} \gamma_{gkh} \right\} \\ &\quad + \left\{ \sum_{k=0}^{c_{hgn}} \pi_{kh} - \sum_{k=0}^{c_{hgn}} \gamma_{gkh} \right\} + \alpha_{hgn} U(c_{hgn} - b_{hgn} - S_{hg}). \end{aligned} \tag{23}$$

The group subproblem can be solved by using DP in a way similar to that presented in Luh *et al.* (1998). Since the run beginning and completion times are all decision variables, each run has two stages, one corresponding to the run beginning time and the other the run completion time. The stagewise costs for a run beginning stage and a run completion stage are given by the first and second terms on the right-hand side of Equation (23), respectively. Since $\alpha_{hgn} U(c_{hgn} - b_{hgn} - S_{hg})$ is associated with the beginning and completion times of the same run, it represents the state transition cost when moving from the run beginning stage to the run completion stage. If this run exists ($c_{hgn} \ge b_{hgn} + S_{hg}$), there will be a penalty of α_{hgn}. Otherwise, there will be no such penalty. With these stagewise and state transition costs, the group subproblems can be solved by using the standard backward DP procedure. The cumulative costs are calculated by moving from the completion stage of the last run to the beginning stage of the last run, and then moving to the completion stage of the preceding run subject to Equation (9). This process then repeats until the beginning stage of the first run is reached. The setup operator type and run beginning and completion times are determined by forward tracing the stages.

4.5. *Dual problem and updating Lagrange multipliers*

4.5.1. *The dual problem*

Let L_l^* denote the minimal lot subproblem cost for lot l and L_{hg}^* the minimal group subproblem cost for group g on machine type h, then the high level dual problem is:

$$\max_{\{\pi_{ko},\pi_{kh},\gamma_{gkh},\lambda_{lj}\}} D,$$

with

$$D \equiv \sum_{l} L_l^* + \sum_{h,g} L_{hg}^* - \sum_{k,o} \pi_{ho} O_{ko} - \sum_{k,h} \pi_{kh} M_{kh} + \sum_{l,j} \lambda_{lj} s_{lj}. \quad (24)$$

4.5.2. *Solving the dual problem*

Among the existing methods for solving dual problems, the subgradient method is the most widely used technique. Since the subgradient method requires the minimization of all subproblems before each update of the multipliers, solving the subproblems becomes time consuming for large problems with many lots and runs. To overcome this difficulty, the Interleaved SubGradient (ISG) method was developed in Kaskavelis and Caramanis (1998) and later extended to the Surrogated SubGradient (SSG) method in Zhao *et al.* (1999) where a proof of convergence is provided. These two methods update the multipliers after solving each subproblem, and converge faster than the subgradient method especially for large problems. In our study, the SSG method is used to solve the dual problem, Equation (24).

4.6. *Heuristics*

The updating of multipliers is stopped after a fixed amount of computation time or a fixed number of iterations have been executed. Since resource capacity constraints and the operation precedence constraints related to completion times have been relaxed, subproblems solutions generally do not constitute a feasible schedule when put together. A heuristic procedure is thus developed to adjust the subproblem solutions to form a feasible schedule following Luh *et al.* (1998) as is summarized next.

A list of operations is first created by arranging all the operations in the ascending order of their beginning times obtained from the optimization. An operation for a lot can be started after its first transfer lot has finished the predecessor operation, the machine has been set up, and the required machine and operator are available. The corresponding operation completion time is calculated based on the lot's beginning time, the processing time of a transfer lot, the time of possible intermittent idling as described by Equation (8), and the availability of machines and operators. If machine or operator capacity constraints are violated at time k, a greedy heuristic based on the incremental change in J determines which operation should begin at that time unit, and which ones should be delayed. Setup for a machine is determined based on the machine's status. If the machine has been set up for a group and the next lot to be processed belongs to the same group, then no setup is needed. Otherwise a setup is needed.

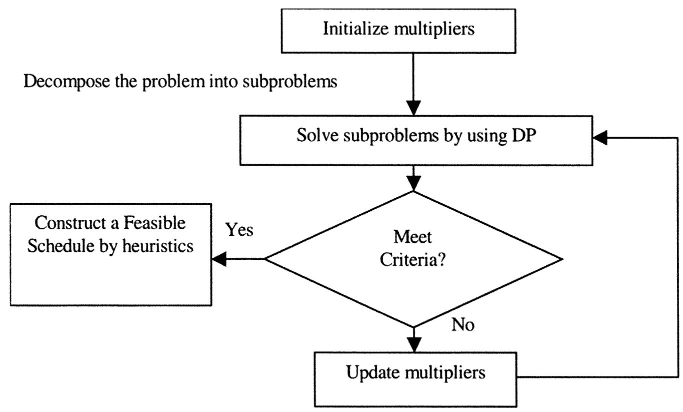

Fig. 3. Flowchart of the solution methodology.

The quality of a feasible schedule obtained is quantitatively evaluated by its relative duality gap, which is the relative difference between the feasible schedule cost J and the largest dual value D obtained, i.e., duality gap $= (J - D)/D \times 100\%$.

The overall solution methodology is summarized in the flowchart displayed in Fig. 3.

5. Numerical results

The method has been implemented by using the object-oriented programming language C++, with a Microsoft Access user interface. Testing has been performed on a Pentium Pro200 PC, and four examples are reported here to demonstrate the performance of the developed method. The first two small examples concentrating on multiple resources and transfer lots are used to present the solutions in detail and the insights obtained. The third example draws on data from J. M. Products Inc. to demonstrate that our method can generate near-optimal schedules within a reasonable computational time for problems of practical sizes. The fourth example is to demonstrate the performance of the method with randomly generated data. For all the examples, multipliers are initialized at zero.

5.1. *Example 1 (Multiple resources)*

This example is to demonstrate the benefit of simultaneously considering both machines and operators in the optimization. In the problem, five lots with one part in each lot are to be scheduled on three different machines attended by three operators. Setup is not needed for any operation, and the data is shown in Table 1.

The problem is first solved by considering the operator capacity constraints, Equation (3), in both the optimization and heuristics as presented in this paper in 2.00 CPU

Table 1. Data for example 1

Lot l	*Op. j*	*Mach.* H_{lj}	t_{ljh}	d_l	w_l	O_{lj}	m_{ljo}
1	1	1	1	0	1	1	1
	2	2	2			1	0.7
	3	2	3			3	1
2	1	3	3	3	1	2	1
	2	1	2			2	1
3	1	3	2	5	1	3	0.6
	2	1	2			1	1
4	1	2	1	10	1.5	1	1
	2	3	2			2	1
	3	3	1			1	1
5	1	1	2	8	0.5	1	0.3
	2	2	1			3	0.5

seconds. The Gantt chart of the schedule obtained is shown in Fig. 4 with a cost of 34. The lower bound D obtained is 33.69 with a relative duality gap 0.92%. For this small example, it can be shown that the schedule is optimal by exhaustive search.

The problem is then solved by ignoring the operator capacity constraints in the optimization and considering them only in the heuristics for the same 2.00 CPU seconds. The Gantt chart of the schedule obtained is shown in Fig. 5 with a cost of 42.

The cost obtained by our method is 23.5% lower than that obtained by the method not considering operator capacity constraints in the optimization. This can be explained as follows. When operator constraints are ignored in the optimization, operation (2, 1) should be scheduled before (3, 1) to avoid a higher tardiness penalty. This sequence, however, fails to consider the fact that both (1, 3) and (3, 1) need operator 3 therefore one of them has to be delayed.

5.2. *Example 2 (Transfer Lot)*

This example is to demonstrate that scheduling with transfer lots can greatly improve the system performance. There are three lots with an equal weight of one to be scheduled on three machine types, with one machine per type. There are four parts in lot 1, and two parts in lots 2 and 3. Setup is not needed and operators are always available. The data is given in Table 2.

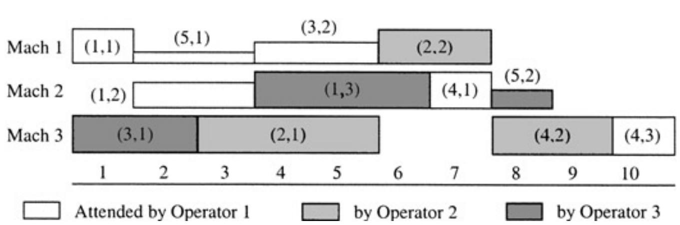

Fig. 4. Schedule considering operator capacity constraints in the optimization.

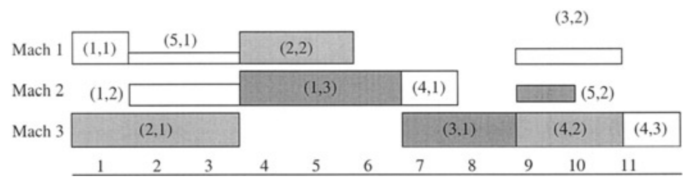

Fig. 5. Schedule ignoring operator capacity constraints in the optimization.

The problem is first solved by treating each part as a transfer lot in 3.00 CPU seconds. The Gantt chart of the schedule obtained is shown in Fig. 6 with a cost of 605. The lower bound is 588.89 with a relative duality gap of 2.74%.

This problem is then solved without considering transfer lots for the same CPU seconds. The Gantt chart of the schedule obtained is shown in Fig. 7 with a cost of 1021. The lower bound is 999.00 with a relative duality gap of 2.21%.

The cost obtained considering transfer lots is 68.7% lower than that without considering transfer lots, implying that transfer lots can significantly improve on-time delivery and reduced inventory.

5.3. *Example 3 (Data from J. M. Products Inc.)*

This example draws on data from J. M. Products Inc. In the example, 32 lots with a total of 12 000 parts belonging to 15 part types are to be scheduled on 14 machine types with a total of 27 machines over a time horizon of 2800. The average number of operations per part is 5.8, and 42.5% of the operations need setups belonging to 53 groups. Setups can only be done by one of three setters, while operations can be attended by either a setter or one of six operation operators. About 27.6% of all operations require partial operator attention. Three cases are discussed below.

The first case is to demonstrate the capability of our method to solve a practical problem. Three subcases are considered without additional penalties on excessive setups:

1. transfer lot size = 100;
2. transfer lot size = 200;
3. no transfer lot considered.

Table 2. Data for example 2

Lot l	*Op. j*	*Mach. H*	t_{ljh}	d_l
1	1	1	2	1
	2	2	1	
	3	3	2	
2	1	1	3	0
	2	2	1	
	3	3	2	
3	1	2	1	1
	2	3	2	
	3	1	3	

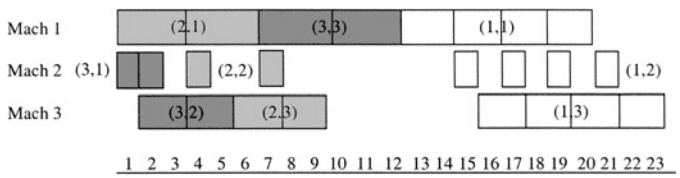

Fig. 6. Schedule considering transfer lots.

For each subcase, the algorithm is terminated after 30 iterations, and the test results are summarized in Table 3. It can be seen that the algorithm generated near-optimal schedules within a reasonable computational time. Schedules considering transfer lots are significantly better than the one without considering transfer lots; and for the former, the one with a smaller transfer lot size has a better solution as expected although it requires a slightly more computational time.

The second case is to demonstrate the effects of having additional penalties on excessive setups in the objective function, Equation (10). The data of the first subcase is used with an additional penalty of 200 on all runs beyond the first one within each group. With 30 iterations, the schedule obtained has 79 setups with a feasible tardiness and earliness cost of 62 263. Compared with the first subcase that has 90 setups and a feasible cost of 52 181, the number of setups in this case is significantly reduced, although the schedule has a higher feasible cost. This example thus demonstrates a trade-off between on-time delivery and a small number of setups.

The third case is to compare our method with a "quick heuristic," and the testing is based on the data of the first case. The "quick heuristic" is as follows. By ignoring machine and operator capacity constraints, the beginning times of all operations of all lots are first determined based on the lot due dates, subject to the operation precedence constraints and arrival time constraints. The heuristics described in Section 4.6 is then used to construct a feasible schedule based on the beginning times thus obtained. Using the "quick heuristic," schedules are obtained in less than 30 seconds for each of the subcases, and have feasible costs of 69 183, 76 379, and 89 815, respectively. They are 32.5, 35.8 and 41.9% higher than the 52 181, 56 232, and 63 270, respectively, obtained by using our method, implying that our method significantly outperforms this "quick heuristic."

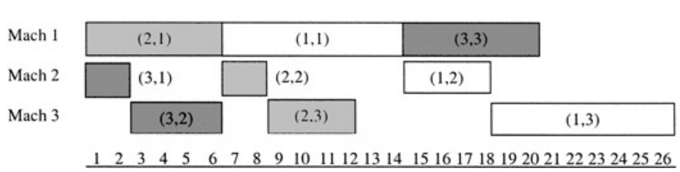

Fig. 7. Schedule without considering transfer lots.

Table 3. Test results for example 3

Subcases	*Transfer lot size*	*Feasible cost*	*Duality gap (%)*	*CPU time (seconds)*
1	Transfer lot size = 100 Number of transfer lots = 120	52 181	8.94	307
2	Transfer lot size = 200 Number of transfer lots = 60	56 232	9.18	299
3	No transfer lot	63 270	6.74	281

5.4. *Example 4 (Data randomly generated)*

This example is to further demonstrate the performance of our method with the data in example 3 randomly modified. Three subcases are considered. The first subcase is to randomly increase or decrease the number of parts in each lot and its due date. The number of parts within each lot has a 50% probability to be increased or decreased by a random integer uniformly distributed over [0, 300], and the due date is changed in a similar way. A lot is then divided into equal-sized transfer lots each having as many parts as possible but less than 150, and the algorithm is terminated after 10 minutes. The procedure is repeated 25 times (25 simulation runs), and the smallest, largest, and average duality gaps obtained in 10 minutes are 5.41, 17.93 and 9.21%, respectively, implying that all schedules are obtained with near-optimal quality in a reasonable computational time. The algorithm is then retested for each run with 15 minutes. The results obtained have a higher dual costs for all 25 simulation runs but have lower feasible costs only for three simulation runs as compared with the results obtained in 10 minutes. Better dual costs are thus obtained as the computation time increases. Good feasible costs, however, can be generally obtained within a reasonable CPU time (10 minutes in the case).

The second case is based on the data of the first subcase of case 1 in example 3 with the number of operators randomly increased by an integer uniformly distributed over [0, 2]. Each new operator has a 50% probability to be a setter or an operation operator, and the algorithm is terminated after 10 minutes. The procedure is repeated 25 times, and the smallest, largest and average duality gaps thus obtained are 4.5, 11.39 and 7.82%, respectively. The number of operators is then randomly decreased in the same way, and the algorithm is terminated in 10 minutes and repeated 25 times. The smallest, largest and average duality gaps thus obtained are 6.47, 16.23 and 8.74%, respectively. Near-optimal schedules are thus obtained in a reasonable computational time.

The third case is based on the data of the first subcase of case 1 in example 3 with additional penalties on excessive runs beyond the first few within each group. The number of runs without additional penalties is randomly determined based on an integer uniformly distributed over [1, 4], and the algorithm is terminated after 10 minutes. The procedure is repeated 25 times, and the average number of setups obtained is 85 with a feasible cost of 57 254 as compared to the first subcase of example 3 having 90 setups and a feasible cost of 52 181. This case thus demonstrates a trade-off between on-time delivery and a small number of setups.

6. Conclusions

The modeling of scheduling features of particular interest to small manufacturers and the mathematical resolution of the resulting problem have been presented. The simultaneous consideration of machines and operators can be extended to solve problems with general multiple resource requirements (machines, operators, tools, pallets, etc.). The indirect consideration of the goal for small numbers of setups effectively achieves the goal without introducing additional variables or major computation requirements. The idea to partially relax precedence constraints is of practical significance, and can be used to handle other cases when complicated operation precedence relations are involved (e.g., job shop scheduling with fuzzy processing times). Extensive numerical testing demonstrates that high quality schedules are obtained for problems of practical sizes within reasonable CPU times.

Acknowledgements

This work was supported in part by the National Science Foundation under Grants DMI-9813176 and also by DMI-0223443, and also by J. M. Products Inc. The authors would like to thank Mr. Jinlin Wang of PeopeleSoft Inc. for his invaluable help in the development of the scheduling software. The authors also would like to thank Professor Haoxun Chen, Dr. Xing Zhao, Dr. Yuanhui Zhang, and Dr. Feng Liu formally at UConn for the insightful discussions in the development and implementation of the transfer lot algorithm.

Nomenclature

a_l	arrival time of lot l;
b_l	beginning time of lot l;
b_{ld}	desired start time of lot l;
b_{lj}	beginning time of operation (l, j);
b_{hgn}	beginning time of run (h, g, n);
c_l	completion time of lot l;
c_{lj}	completion time of operation (l, j);
c_{hgn}	completion time of run (h, g, n);
d_l	due date of lot l;
D	dual cost;
E_l	earliness of lot l;
h	machine type index;
h_{lj}	selected machine type index for (l, j);
H	set of machine types;
H_{lj}	set of machine types eligible for (l, j);
$\lvert H_{lj}\rvert$	cardinality of H_{lj};
(h, g, n)	the nth run of group g on machine type h;
l	lot index $1 \leq l \leq L$;
(l, j)	the jth operation of lot l;
j	operation index, $1 \leq j \leq J_l$;
J	feasible cost;
J_l	number of operations of lot l;
k	time index, $0 \leq k \leq K - 1$;
K	time horizon of scheduling;
L	total number of lots to be scheduled;
L	Lagrangian function;
L_l	lot subproblem of lot l;
L_{hg}	group subproblem of group g on machine type h;
L^*_{hg}	optimal cost for group g subproblem on machine type h;
L^*_l	optimal cost for subproblem of lot l;
m_{ljo}	percentage of type o operator attention for operation (l, j);
M_{kh}	number of type h machines available at time k;
N_l	number of transfer lots in lot l;
o	operator type index;
o_{lj}	selected operator type index for (l, j);
o_{hgn}	selected operator type index for (h, g, n);
O	set of operator types;
O_{lj}	set of operator types eligible for (l, j);
$\lvert O_{lj}\rvert$	cardinality of O_{ij};
O_{hg}	set of operator types eligible for setup of group g on machine type h;
$\lvert O_{hg}\rvert$	cardinality of O_{hg};
O_{ko}	number of type o operator available at time k;
S	set of all operations requiring setup;
S_{hg}	setup time of group g on machine type h;
s_{lj}	required "time-out" between operation $(l, j-1)$ and (l, j);
$\bar{S}$	set of all operations that do not need setup;
t_{ljh}	processing time of operation (l, j) on machine type h for a transfer lot;
T_l	tardiness of lot l;
$U(x)$	unit step function $$U(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise;} \end{cases}$$
w_l	weight of tardiness penalty for lot l;
α_{hgn}	penalty for run (h, g, n);
β_l	weight of earliness penalty for lot l;
δ_{ljkh}	0-1 operation variable $$\delta_{ljkh} = \begin{cases} 1 & \text{if } (l, j) \text{ is active on machine type } h \text{ at time } k, \\ 0 & \text{otherwise;} \end{cases}$$

δ_{ljko} 0-1 operation variable

$$\delta_{ljko} = \begin{cases} 1 & \text{if } (l, j) \text{ is performed by} \\ & \quad \text{type } o \text{ operator at time } k, \\ 0 & \text{otherwise;} \end{cases}$$

φ_{khgn} 0-1 run variable

$$\varphi_{khgn} = \begin{cases} 1 & \text{if run } (h, g, n) \text{ is active at time } k, \\ 0 & \text{otherwise;} \end{cases}$$

ζ_{khgn} 0-1 setup variable

$$\zeta_{khgn} = \begin{cases} 1 & \text{if } (h, g, n) \text{ is being set up on time} \\ & \quad k \text{ belonging to time interval} \\ & \quad [b_{hgn}, (b_{hgn} + S_{hg} - 1)], \\ 0 & \text{otherwise;} \end{cases}$$

ζ_{kogn} 0-1 setup variable

$$\zeta_{kogn} = \begin{cases} 1 & \text{if run } n \text{ of group } g \text{ is being set} \\ & \quad \text{up by type } o \text{ operator at time} \\ & \quad k \text{ belonging to time interval} \\ & \quad [b_{hgn}, (b_{hgn} + S_{hg} - 1)], \\ 0 & \text{otherwise;} \end{cases}$$

π_{kh} Lagrangian multiplier of machine type h at time k;

π_{ko} Lagrangian multiplier of operator type o at time k;

γ_{gkh} Lagrangian multiplier of group g on machine type h at time k;

λ_{lj} Lagrangian multiplier for precedence constraints related to completion times;

Δ_{lj} 0-1 integer variable distinguishing if (l, j) is the first operation of lot l.

References

Brucker, P. and Thiele, O. (1996) A branch and bound method for the general-shop problem with sequence dependent setup-times. *OR-Spectrum*, **18**(3), 145–161.

Chen, T. and Hsia, T.C. (1994) Job shop scheduling with multiple resources and an application to a semiconductor testing facility, in *Proceedings of the 33rd IEEE Conference on Decision and Control*, IEEE, Lake Buena Vista, FL, pp. 1564–1570.

Czerwinski, C. and Luh, P.B. (1994) Scheduling products with bills of materials using an improved Lagrangian relaxation technique. *IEEE Transactions on Robotics and Automation*, **10**(2), 99–111.

Dobson, G. and Karmarkar, U.S. (1989) Simultaneous resources scheduling to minimize weighted flow times. *Operations Research*, **37**(4), 592–600.

Dobson, G. and Karmarkar, U.S. (1993) Large scale shop scheduling: formulations and decompositions. Presented at the LAAS Paris Workshop on Production Planning and Scheduling, Toulouse, France.

Gargeya, V.B. and Deane, R.H. (1996) Scheduling research in multiple resource constrained job shops: a review and critique. *International Journal of Production Research*, **34**(8), 2077–2097.

Kaskavelis, C.A. and Caramanis, M.C. (1998) Efficient Lagrangian relaxation algorithms for real-life-size job-shop scheduling problems. *IIE Transactions*, **30**(11), 1085–1097.

Kim, S.C. and Bobrowski, P.M. (1994) Impact of sequence-dependent setup time on job shop scheduling performance. *International Journal of Production Research*, **32**(7), 1503–1520.

Liao, C. and Chuang, C. (1996) Sequencing with setup time and order tardiness trade-offs. *Naval Research Logistics*, **43**(7), 971–984.

Liu, G. and Luh, P.B. (1996) Scheduling job shops with transfer lots, in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, IEEE, Piscataway, NJ, pp. 2523–2528.

Luh, P.B., Liu, F. and Moser, B. (1999) Scheduling of design projects with uncertain number of iterations. *European Journal of Operational Research*, **113**, 575–592.

Luh, P.B., Gou, L., Zhang, Y., Nagahora, T., Tsuji, M., Yoneda, K., Hasegawa, T., Kyoya, Y. and Kano, T. (1998) Job shop scheduling with group-dependent setups, finite buffers, and long time horizon. *Annals of Operations Research*, **76**, 233–259.

Ovacik, I.M. and Uzsoy, R. (1994) Rolling horizon algorithm for a single-machine dynamic scheduling problem with sequence-dependent setup times. *International Journal of Production Research*, **32**(6), 1243–1263.

Panwalkar, S.S., Dudek, R.A. and Smith, M.L. (1973) Sequencing research and the industrial scheduling problem, in *Proceedings of the Symposium on the Theory of Scheduling and its Applications*, pp. 29–32.

Soumen, G. and Cheryl, G. (1993) Production scheduling in a flexible manufacturing system with setups, *IIE Transactions*, **25**(5), 21–35.

Treleven, M.D. and Elvers, D.A. (1985) An investigation of labor assignment rules in a dual-constrained job shop. *Journal of Operations Management*, **6**, 51–68.

Trietsch, D. and Baker, K.R. (1993) Basic techniques for lot streaming. *Operations Research*, **41**(6), 1065–1076.

Vickson, R.G. and Alfredsson, B.E. (1992) Two- and three-machine flow shop scheduling problems with equal sized transfer batches. *International Journal of Production Research*, **30**(7), 1551–1574.

Young, H.L., Bhaskaran, K. and Pinedo, M. (1997) A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Transactions*, **29**(1), 45–52.

Zhao, X., Luh, P.B. and Wang, J. (1999) Surrogate gradient algorithm for Lagrangian relaxation. *Journal of Optimization Theory and Application*, **100**(3), 699–712.

Biographies

Dong Chen received a B.S. degree in Mathematics and an M.S. degree in Computer and Systems Sciences from Nankai University, China, in 1984 and 1987, respectively. He received his Ph.D. degree in the Department of Electrical and Computer Engineering, University of Connecticut. From 1987 to 1990, he was an Assistant Professor in the Department of Computer and Systems Sciences, Nankai University. From 1990 to 1995, he was an Assistant Professor and Associate Professor in the Department of Finance, Dongbei University of Finance and Economics, China. During 1999–2001, he was a consultant in i2 Technologies Inc. He then joined Imagine Software Inc as a quantitative developer. Currently he is with Aristeia Capital as a senior developer. His major interests include manufacturing scheduling, optimizaton, financial products, and software design and implementation. He received the Science and Technology Award honored by the National Education Commission, China, and the Best Student Paper Award at the First International Conference on Quantitative Management, 1997.

Peter B. Luh received his B.S. degree in Electrical Engineering from the National Taiwan University, Taipei, Taiwan, Republic of China in 1973, an M.S. degree in Aeronautics and Astronautics Engineering from MIT, Cambridge, MA in 1977, and a Ph.D. degree from Harvard University in 1980. Since 1980 he has been with the University of Connecticut, and is currently the SNET Professor of Communications & Information Technologies in the Department of Electrical and Computer Engineering, the Director of the Taylor L. Booth Engineering Center for Advanced Technology at the University of Connecticut, and a Visiting Professor of Tsinghua University, Beijing, China. He is interested in planning, scheduling, and coordination of design, manufacturing, and supply chain activities; and schedule, bid, and portfolio optimization and load/price forecasting for power systems. He is a Fellow of IEEE, Editor-in-Chief of *IEEE Transactions on Robotics and Automation*, the founding Editor-in-Chief of the newly created *IEEE Transactions on Automation Sciences and Engineering*, an Associate Editor of *IIE Transactions on Design and Manufacturing*, and an Associate Editor of *Discrete Event Dynamic Systems*.

Lakshman Thakur received his Eng.Sc.D. from Columbia University in 1971 and his B.Sc. from Bombay University in 1963. He has previously been a Visiting Professor of Operations Research at the Yale School of Management (1985–1987). His primary research interests are in the development and applications of linear, nonlinear, and integer programming methods in Management Science and function approximations in optimization mathematics. He has published in many journals, including *Management Science*, *Mathematics of Operations Research*, *SIAM Journal on Applied Mathematics*, *SIAM Journal on Optimization and Control*, *IEEE Transactions*, *Journal of Mathematical Analysis and Applications*, and *Naval Research Logistics*. He has served as a consultant to the IBM Corporation on their Manpower Planning with Risk Assessment System and New Product Warranty System as well as a senior consultant and Director of Management Science in a consulting organization. He is an Associate Editor of *Naval Research Logistics*. His current research focuses on production scheduling, product design and facility location problems. His research on scheduling (with Dr. P. B. Luh) is supported by a National Science Foundation grant. He is an Ackerman Scholar in the Business School, Co-Director Management & Engineering for Manufacturing (MEM), University of Connecticut, and a member of the Operations Research Society of America, The Institute of Management Sciences, and Mathematical Programming Society.

Jack Moreno Jr. is the president of J. M. Products Inc which is a small manufacturer located in Southbury, Connecticut.