



A GA methodology for the scheduling of yarn-dyed textile production

Hsi-Mei Hsu^{a,*}, Yai Hsiung^{b,1}, Ying-Zhi Chen^{a,2}, Muh-Cherng Wu^{a,3}

^a Department of Industrial Engineering and Management, National Chiao Tung University, Hsin-Chu, Taiwan, ROC

^b Department of Information Management, Ta Hwa Institute of Technology, Hsin-Chu, Taiwan, ROC

ARTICLE INFO

Keywords:

Scheduling
Sequence-dependent setup
Multi-stage
Textile
Genetic algorithm
Group-delivery

ABSTRACT

This paper presents a scheduling approach for yarn-dyed textile manufacturing. The scheduling problem is distinct in having four characteristics: multi-stage production, sequence-dependent setup times, hierarchical product structure, and group-delivery (a group of jobs pertaining to a particular customer order must be delivered together), which are seldom addressed as a whole in literature. The scheduling objective is to minimize the total tardiness of customer orders. The problem is formulated as a mixed integer programming (MIP) model, which is computationally extensive. To reduce the problem complexity, we decomposed the scheduling problem into a sequence of sub-problems. Each sub-problem is solved by a genetic algorithm (GA), and an iteration of solving the whole sequence of sub-problems is repeated until a satisfactory solution has been obtained. Numerical experiment results indicated that the proposed approach significantly outperforms the EDD (earliest due date) scheduling method—currently used in the yarn-dyed textile industry.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Yarn-dyed textiles are distinct in their manufacturing processes in which yarn must be dyed before weaving, while most other textiles are first woven and then dyed. A yarn-dyed textile product, for example a shirt, contains several *patterns cloths*. A pattern cloth manifests itself by a particular pattern of colors. In a colorful shirt, its sleeve may be a single-color pattern while its pocket may be a three-color pattern. A three-color pattern is composed of three different color *yarns*, with each color yarn being individually dyed. Only when the three different color yarns have been dyed, they could be woven into the three-color pattern cloth.

Group-delivery is an essential characteristic in the dyeing process. Referring to the shirt shown in Fig. 1, we have five different color yarns to be dyed in the dyeing stage. To weave each pattern cloth, all its composing yarns have to be delivered to the weaving machine in a *group* manner. That is, only when all the composing yarns of a particular pattern cloth arrive at the weaving machine, can the weaving of the pattern cloth be carried out.

Likewise, *group-delivery* is also an essential characteristic in the weaving process. See the shirt shown in Fig. 1, we have three pattern-cloths to be woven. For effectively making the shirt, the three

pattern-cloths also have to be delivered in a group manner. That is, only when all the three pattern-cloths have shipped to the downstream shirt-maker, can the shirt-maker start to manufacture the shirt.

In addition, the dyeing process is distinct in having a *setup dependency* characteristic. Before dyeing a yarn, we need to clean the dyeing tank—the machine that processes the yarn to be dyed. The clean time (setup time) required to prepare for dyeing a coming job can be different, dependent upon the colors of the coming yarn and the one just finishing dyeing. Consider two consecutive dyeing jobs. If the preceding job is dark-color (e.g. black) and the following one is light-color (e.g. yellow), then we need a *thorough cleaning* for the dyeing tank. That is, before dyeing the light-color job, the dark-coloring agent in the tank should be completely removed. In contrast, if the preceding job is light-color and the following one is dark-color, then we need only a *rough cleaning* for the dyeing tank. The time required for a thorough cleaning is much longer than that for a rough cleaning. This feature indicates that the dyeing process is *sequence-dependent* in setup time.

In summary, the manufacturing of the yarn-dyed textiles essentially involves two consecutive production processes—dyeing and weaving. These two processes are distinct in three points: (1) group-delivery in the dyeing process, (2) group-delivery in the weaving process, and (3) sequence-dependent in the dyeing process. To our knowledge, scheduling problems concerning these three features as a whole have not been examined in literature.

This paper formulated the scheduling problem for the yarn-dyed textile manufacturing process as a mixed integer program, and developed a genetic algorithm based approach to solve the

* Corresponding author. Tel.: +886 3 5731761.

E-mail addresses: hsimei@cc.nctu.edu.tw (H.-M. Hsu), yai@thit.edu.tw (Y. Hsiung), yzchen@mail.nctu.edu.tw (Y.-Z. Chen), mcwu@mail.nctu.edu.tw (M.-C. Wu).

¹ Tel.: +886 3 5927700x2751; fax: +886 3 5923957.

² Tel.: +886 3 5731761.

³ Tel.: +886 3 5731913.

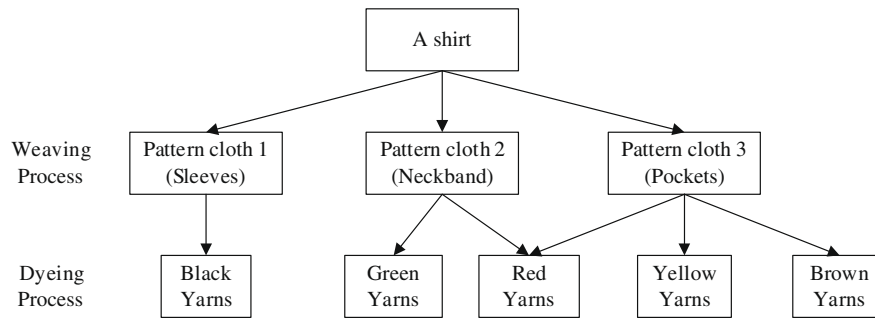


Fig. 1. Hierarchical product structure of a shirt example.

scheduling problem. Numerical experiments indicated that the proposed GA method significantly outperforms the EDD (earliest due date) scheduling method, which is most commonly used in the yarn-dyed textile industry.

The remainder of this paper is organized as follows: Section 2 gives a review on relevant literature. Section 3 explains the scheduling problem in more detail. Section 4 presents a mathematical formulation of the scheduling problem. Section 5 describes the solution architecture and the genetic algorithm (GA) used in each module of the architecture is presented in Section 6. Numerical experiments are presented in Section 7 and concluding remarks are in the last section.

2. Literature review

Our scheduling problem has three distinct features: *multiple-stage production*, *sequence-dependent setup times*, and *group-delivery*. Relevant literature is reviewed below and categorized into two groups: scheduling on textile production, and scheduling with sequence-dependent setup times.

2.1. Scheduling on textile production

Numerous studies on the scheduling of textile manufacturing have been published. Some addressed single-stage systems, and some others addressed multiple-stage systems.

Numerous textile scheduling studies on a single-stage system have been published. Tang, Hammond, and Abernathy (1994) propose a scheduling model for an apparel production system in order to allocate production capacity and schedules jobs for each production line. Saydam and Cooper (1995) develop a computer-based system for the scheduling of dyeing textile fabrics in order to maximize the machine utilization. Ford and Rager (1995) develop an expert system to aid the design of textile manufacturing process. Shiroma and Niemeyer (1998) present a scheduling method, for a textile company with multiple factories, in order to distribute and schedule production jobs among the factories. Wong, Chan, and Ip (2000) present a method to solve the job sequencing problem in the garment industry, in particular for spreading and cutting machines.

Some other literature studies the scheduling on multiple-stage textile systems. Sun and Chisman (1991) develop a simulation model to assist production scheduling for a multiple-stage textile-belt manufacturing process. Tomastik, Luh, and Liu (1996) present a scheduling approach especially for cellular manufacturing in apparel industry, in order to determine when to set up a cell and when to release garments into the cell. Min and Cheng (2006) address a textile scheduling problem, in which the due date is a decision variable; and they attempt to find an optimal due date as well as an optimal schedule in order to minimize the total cost. Guo, Wong, Leung, Fan, and Chan (2006) deal with a scheduling

problem for apparel manufacturing by formulating it as a mixed integer programming model and use genetic algorithm to solve the problem.

Many scheduling studies on multiple-stage textile systems have been published. Yet, only a few address the effects of sequence-dependent setup times. For example, Karacapilidis and Pappis (1996) addressed the scheduling for a textile manufacturing process with a sequence-dependent setup feature; however, their textile products are not yarn-dyed. That is, yarns are weaved before dyed in their process. Such a weaving-before-dyeing process simplifies the dyeing tasks, and therefore does not have the group-delivery feature.

2.2. Scheduling with sequence-dependent setup time

Scheduling problems with *sequence-dependent* setup times have been widely investigated, with a few survey papers having been available (Allahverdi, Gupta, & Aldowaisan, 1999; Cheng, Gupta, & Wang, 2000; Yang & Liao, 1999).

Some of these studies examine the scheduling of a single-stage production system. They are varied in dealing with a single objective function (Luo & Chu, 2006; Wang & Wang, 1997), with a multiple-objective function (Lee & Asllani, 2004), or with parallel-machines in the production system (Dastidar & Nagi, 2005).

Some others study the scheduling with *sequence-dependent* setup for a multiple-stage production system. Examples solution approaches to these studies include the use of integer programming (Liu, 1996), the use of immune algorithm (Zandieh, Ghomi, & Hussein, 2006), and the use of genetic algorithm (Ruiz & Maroto, 2006; Ruiz, Maroto, & Alcaraz, 2005).

Noticeably, most prior scheduling studies with sequence-dependent setup time have an implicit assumption: all the jobs to be scheduled are independent in their due assignment. This implies that all the scheduled jobs have no *group-delivery* feature. That is, there does not exist a BOM (bill of materials) to interrelate these jobs.

In summary, for the three scheduling features—multiple-stage production, sequence-dependent setup time, and group-delivery, previous studies only addressed some of them partially. Our research is unique in addressing the three scheduling features as a whole.

3. Production process of yarn-dyed textile

As shown in Fig. 2, the manufacturing process of yarn-dyed textiles involves three major stages. Yarns are firstly colored in the dyeing stage; subsequently through a stage for treatment/starching; and are finally sent to the weaving stage to make cloth patterns. With relatively lower equipment costs, the capacity of treatment/starching stage is typically equipped much higher than demand; and the production cycle time is a constant. We therefore

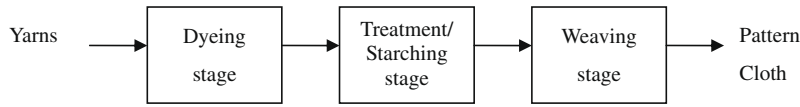


Fig. 2. Production process of yarn-dyed.

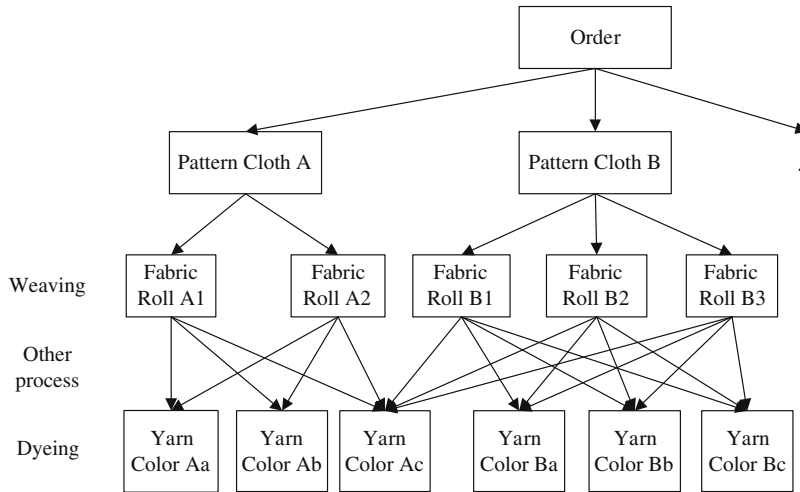


Fig. 3. BOM of an.

consider only the dyeing and the weaving stages in investigating the scheduling problems on yarn-dyed textiles.

3.1. Hierarchical structure of an order

Apparel companies are the downstream customers for a yarn-dyed textile manufacturer. While receiving a customer order, the yarn-dyed manufacturer has to formulate a particular BOM (bill of materials) for the order. As shown in Fig. 3, such an order may involve several *pattern-cloths* for making apparels; a pattern cloth is composed of several *fabric rolls*; and a fabric roll contains several different *colored yarns*. For each colored yarn used in an order, only one dyeing job can be designated to ensure color uniformity. A dyeing job therefore may support more than one fabric rolls. Fig. 3—an elaboration of Fig. 1 emphasized that a weaving task, if with a large volume, is decomposed into several smaller weaving jobs (each job is called a *fabric roll*) in order to reduce the weaving cycle time.

Noticeably, for a pattern cloth, each of its fabric rolls can be *independently* processed on a weaving machine. However, their dyeing tasks have to be processed *dependently* or so called in a batch manner. That is, the colored yarns to be used for different fabric rolls for a particular pattern cloth have to be dyed on the same machine in order to maintain color consistency. For example, fabric rolls A1 and A2 can be weaved on different machines; but each of their component yarns (e.g. color Aa) has to be dyed on the same machine.

3.2. Group-delivery characteristics

Group-delivery is an essential feature in the weaving process. In weaving process, an order’s on-time delivery is determined by a group of fabric rolls. See Fig. 3, a customer order, involving several pattern cloths, is in turn composed of several fabric rolls. Only when all the fabric rolls are completed, can all the pattern-cloths of an order be delivered to the customer. This group-delivery

requirement implies that the latest completion time of fabric rolls is critical to an order’s on-time delivery.

The group-delivery characteristic also appears in the dyeing process. A fabric roll—a weaving job, for producing a particular pattern cloth, may involve several different colored yarns, each of which should be produced in one dyeing job to keep color uniformity. To weave a fabric roll, all its associated dyeing jobs have to be completed in advance. That is, all the dyeing jobs of a particular fabric roll must be delivered in a group manner.

3.3. Jobs and machines

In the weaving process, there are numerous parallel weaving machines. A fabric roll, a weaving job, can be processed by any of the parallel machines. Once a weaving job is started on a machine, it cannot be preempted until it is completed. Compared to the dyeing process, the weaving process for a typical job is relatively longer in production time. The duration for weaving a fabric roll is around 10 days in general.

In the dyeing process, a particular color of yarn represents a production job. A dyeing machine is a container. To dye a job, raw yarns are put into the container filled with color liquid and stay for a period of time, about 8 h typically. Based on the container size, dyeing machines are classified into four types: small, medium, large, and extra large. Each of the four types may be equipped with several identical dyeing machines.

Some assumptions and constraints about the dyeing process are presented. Before proceeding the scheduling of the dyeing process, we assume that each dyeing job has been assigned to a particular type of machine. Of the machines in the assigned type, exactly which one to use is left to decision. Noticeably, a dyeing job can only be processed on a single machine; that is, the job cannot be shared by two or more dyeing machines.

Dyeing different jobs on a particular dyeing machine needs a setup, which is essentially for cleaning the container. The setup time is sequence-dependent. For example, dyeing a dark-color

job followed by a light-color one needs more setup efforts. This is to ensure the complete removal of dark-pigment residual in order not to contaminate the light-color job. In contrast, dyeing a light-color job followed by a dark-color one needs less setup efforts—because a few amount light-pigment residual would not hurt the dyeing quality of the dark-color job.

4. Problem formulation

The scheduling problem is formulated as a mixed integer programming (MIP) model. The objective is to determine the starting time of each job at the dyeing and weaving stages for minimizing total tardiness of orders. The problem decisions involve assignment of jobs to machines and sequence of jobs for each machine. Wang and Wang (1997) have proposed a 0–1 mixed integer program for single-machine scheduling with sequence-dependent setup time. We extend their model by additionally considering the group-delivery requirement and expand the application domain to a multi-stage scenario. The problem is formulated using the following notations.

Indices

i	order index ($1 \leq i \leq J$)
k	production stage index; $k = d$ represents dyeing stage; and $k = w$ represents weaving stage
J_k^i	production job index ($1 \leq J_k^i \leq J_k^i$)
g_k	machine type index ($1 \leq g_k \leq G_k$)
g_k^m	m th machine in machine type g_k ($1 \leq g_k^m \leq g_k^m$)

Sets

$J(g_k)$	a set of jobs which can be processed by machine type g_k
$J\left(\begin{smallmatrix} i \\ k \end{smallmatrix}\right)$	a set of jobs which belong to order i in stage k
$Com(j_w^i)$	the dyeing components of a weaving job j_w^i —a set of dyeing jobs which must be completed before performing a weaving job j_w^i

Parameters

J_k^i	total number jobs of order i in stage k
$p_{j_k^i, g_k^m}^i$	processing time of job J_k^i at machine g_k^m
o_i	due date of order i
$s_{j_w^i, g_w^m}^i$	time required for setting up a weaving machine g_w^m for processing job J_w^i
$s_{(j'-j)g_d^m}^i$	time required for setting up a dyeing machine g_d^m to process job j while job j is scheduled immediately behind job j' ; ($j' \neq j$)
$l_{j_d^i \rightarrow w}^i$	traveling time required for a dyeing job J_d^i which moves from dyeing stage to weaving stage

Decision variables

$\pi_{j_k^i, g_k^m}^i$	$\begin{cases} 1, & \text{if job } J_k^i \text{ is assigned to machine } g_k^m \\ 0, & \text{otherwise} \end{cases}$
$\phi_{(j'-j)g_k^m}^i$	$\begin{cases} 1, & \text{if job } j \text{ is scheduled immediately behind job } j' \\ & \text{at machine } g_k^m \\ 0, & \text{otherwise.} \end{cases}$

Intermediate variables

${}^q r_{j_w^i, g_w^m}^i$	the earliest starting time for a weaving job J_w^i at machine g_w^m , by considering the availability of machine g_w^m
${}^c r_{j_w^i, g_w^m}^i$	the earliest starting time for a weaving job J_w^i at machine g_w^m , by considering the availability of dyeing components, $Com(j_w^i)$
$r_{j_k^i, g_k^m}^i$	starting time for job J_k^i at machine g_k^m
$f_{j_k^i, g_k^m}^i$	completion time for job J_k^i at machine g_k^m
$\psi_{(j'-j)g_k^m}^i$	$\begin{cases} 1, & \text{if job } j' \text{ is scheduled ahead of job } j \text{ at } g_k^m \\ & \text{(notice : } j' \text{ is not necessarily immediately ahead of } j) \\ 0, & \text{otherwise} \end{cases}$
$J(g_k^m)$	a set of jobs which are assigned to machine g_k^m

To formulate the scheduling problem, two fictitious jobs u and v are introduced (Wang & Wang, 1997). The two jobs are both with zero processing/setup times; u must be firstly processed, and v must be lastly processed. That is, $s_{(u-j)g_k^m} = s_{(j-v)g_k^m} = 0$, $p_{u, g_k^m} = p_{v, g_k^m} = 0$, $r_{u, g_k^m} = f_{u, g_k^m} = 0$. The scheduling problem can be formulated as follows:

$$\text{Min } Z = \sum_{i=1}^I \text{Max}_{j_w^i \in J_w^i} \left\{ \text{Max}[0, (f_{j_w^i, g_w^m}^i - o_i)] \right\} \quad (1)$$

$$\text{s.t. } \sum_{m=1}^M \pi_{j_k^i, g_k^m}^i = 1, \quad \forall j_k^i \in J(g_k), \quad \forall g_k \quad (2)$$

$$\sum_{j' \in J(g_k^m) \cup \{u\}, j' \neq j} \phi_{(j'-j)g_k^m}^i = 1, \quad \forall j \in J(g_k^m) \cup \{v\}, \quad \forall g_k^m \quad (3)$$

$$\sum_{j' \in J(g_k^m) \cup \{v\}, j' \neq j} \phi_{(j'-j)g_k^m}^i = 1, \quad \forall j' \in J(g_k^m) \cup \{u\}, \quad \forall g_k^m \quad (4)$$

$$f_{j_d^i, g_d^m}^i = \left[\sum_{h \in J(g_d^m) \cup \{u\}} \psi_{(h-j)g_d^m}^i \left(\sum_{j' \in J(g_d^m) \cup \{u\}, j' \neq h} s_{(j'-h)g_d^m}^i \cdot \phi_{(j'-h)g_d^m}^i + p_{j_d^i, g_d^m}^i \right) \right] + \left(\sum_{j' \in J(g_d^m) \cup \{u\}, j' \neq j} s_{(j'-j)g_d^m}^i \cdot \phi_{(j'-j)g_d^m}^i + p_{j_d^i, g_d^m}^i \right), \quad \forall j \in J(g_d^m) \cup \{v\}, \quad \forall g_d^m \text{ (where } j = j_d^i) \quad (5)$$

$${}^q r_{j_w^i, g_w^m}^i = \sum_{j' \in J(g_w^m) \cup \{u\}, j' \neq j} \phi_{(j'-j)g_w^m}^i \cdot f_{j_w^i, g_w^m}^i, \quad \forall j \in J(g_w^m) \cup \{v\}, \quad \forall g_w^m \text{ (where } j = j_w^i) \quad (6)$$

$${}^c r_{j_w^i, g_w^m}^i = \text{Max}_{j_d^i \in Com(j_w^i)} (f_{j_d^i, g_d^m}^i + l_{j_d^i \rightarrow w}^i), \quad \forall j_w^i \quad (7)$$

$$r_{j_k^i, g_k^m}^i = \text{Max}({}^q r_{j_k^i, g_k^m}^i, {}^c r_{j_k^i, g_k^m}^i), \quad \forall j_k^i \quad (8)$$

$$f_{j_k^i, g_k^m}^i = r_{j_k^i, g_k^m}^i + (s_{j_k^i, g_k^m}^i + p_{j_k^i, g_k^m}^i), \quad \forall j_k^i \quad (9)$$

$$\pi_{j_k^i, g_k^m}^i, \quad \phi_{(j'-j)g_k^m}^i, \quad \psi_{(j'-j)g_k^m}^i \in \{0, 1\} \quad (10)$$

Eq. (1) states that the objective function is to minimize the total tardiness of orders, where $\text{Max}_{j_w \in J(i_w)} \{ \text{Max}[0, (f_{j_w g_w}^m - o_i)] \}$ denotes that the tardiness of each order i .

Eq. (2) constrains that each job should be only assigned to one machine, for both dyeing and weaving stages. Eqs. (3) and (4) ensure that each job has only one predecessor job, and has only one successor job. Eq. (5) computes the completion time for dyeing job j , where the first term denotes the time epoch when the machine for processing job j becomes available, and the second term denotes the setup/processing times for job j .

Eq. (6) computes the time epoch when the machine for processing weaving job j is available. Eq. (7) computes the time epoch when the dyeing components of weaving job j has been ready for weaving. Eqs. (6) and (7) ensure that the starting time of a weaving job must be later than the ready times of its dyeing components as well as its processing machine. Eqs. (8) and (9), respectively, determines the starting time and the completion time of the weaving jobs. Eq. (10) declares the binary variables.

5. Solution architecture

The addressed scheduling problem is essentially NP-hard in complexity. To reduce problem complexity, we decompose the scheduling problem into two sub-problems. Each sub-problem is solved by a genetic algorithm (GA). The ultimate solution for the scheduling problem is obtained by solving the two sub-problems in an iterative manner.

The decomposition of the scheduling problem is shown in Fig. 4, where the first sub-problem is called weaving module (WM), and the second one is called dyeing module (DM). The scheduling problem is solved in an iterative manner. In a particular iteration, the solution of WM is fed to DM to obtain a solution for the second sub-problem, which in turn is fed back to WM to obtain a new solution for the first sub-problem in the next iteration. The iterative procedure is repeated until a satisfactory solution is obtained.

The procedure for iteratively solving the two sub-problems, called **Weaving Dyeing Scheduling**, is presented below with its notation firstly introduced.

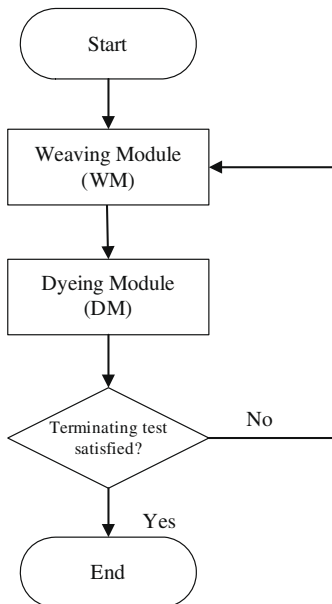


Fig. 4. Solution architecture.

${}^n WMZ$	total tardiness of weaving jobs obtained in n th iteration in WM
${}^n DMZ$	total tardiness of dyeing jobs obtained in n th iteration in DM
n_T	predefined maximum number of iterations
h	predefined threshold value for terminating the iterative procedure
${}^n WM r_{j_w}^i$	planned starting time of weaving job j_w^i obtained in n th iteration
${}^n WM E_{j_w}$	earliest starting time of weaving job j_w^i obtained in n th iteration
${}^n WM f_{j_w}$	completion time of weaving job j_w^i obtained in n th iteration
${}^n DM r_{j_d}^i$	planned starting time of dyeing job j_d^i obtained in n th iteration
${}^n DM E_{j_d}$	earliest starting time of dyeing job j_d^i obtained in n th iteration
${}^n DM f_{j_d}$	completion time of dyeing job j_d^i obtained in n th iteration
$l_{j_d-w}^i$	time required for dyeing job j_d^i moving from dyeing to weaving stages
o_i	due date of order i
${}^n DM O_{j_d}$	planned due date of dyeing job j_d^i obtained in n th iteration
$f_1({}^n WM E_{j_w}, o_i)$	GA solution obtained in the WM module in n th iteration
$f_2({}^n DM E_{j_d}, {}^n DM O_{j_d})$	GA solution obtained in the DM module in n th iteration
$g_1({}^n WM r_{j_w}^i) = {}^n DM O_{j_d} : g_1$	is a procedure that computes ${}^n DM O_{j_d}$ from ${}^n WM r_{j_w}^i$
$g_2({}^n DM f_{j_d}) = {}^n WM E_{j_w} : g_2$	is a procedure that computes ${}^n WM E_{j_w}$ from ${}^n DM f_{j_d}$

Procedure Weaving Dyeing Scheduling

Initiation

$n = 1$;
 Termination = 'No';
 ${}^n WM E_{j_w} = 0$ for each weaving job j_w^i in each order i ;
 ${}^n DM E_{j_d} = 0$ for each dyeing job j_d^i in each order i ;

While (Termination = 'No') Do

$({}^n WMZ, {}^n WM f_{j_w}, {}^n WM r_{j_w}^i) = f_1({}^n WM E_{j_w}, o_i)$; /*Solving Module WM*/
 ${}^n DM O_{j_d} = g_1({}^n WM r_{j_w}^i)$; /*Interface from WM to DM*/
 $({}^n DMZ, {}^n DM f_{j_d}, {}^n DM r_{j_d}^i) = f_2({}^n DM E_{j_d}, {}^n DM O_{j_d})$; /*Solving Module DM*/

${}^n WM E_{j_w} = g_2({}^n DM f_{j_d})$; /*Interface from DM to WM*/

/* Check Termination Status */

If $n = n_T$, then Termination = 'Yes'

Elseif $n \geq 2$ and $\text{Max}_{i, j_w} | {}^n WM E_{j_w} - {}^{n-1} WM E_{j_w} | \leq h$ then Termination = 'Yes';

Else $n = n + 1$; /*repeat iteration*/

Endif

Endwhile

In the iteration loop, the first statement $({}^n WMZ, {}^n WM f_{j_w}, {}^n WM r_{j_w}^i) = f_1({}^n WM E_{j_w}, o_i)$ denotes the input/output of the GA in the WM mod-

ule. Given ${}^{\text{WM}}E_{r_{j_w}^i}$ and o_i , the GA in WM module determines a near-optimal weaving job sequence (represented by job starting time ${}^{\text{WM}}r_{j_w}^i$), and in turn computes ${}^{\text{WM}}f_{j_w}^i$ and $({}^{\text{WM}}Z)$ of the job sequence. Notice that the GA has to ensure ${}^{\text{WM}}r_{j_w}^i \geq {}^{\text{WM}}E_{r_{j_w}^i}$. This means that the starting time (${}^{\text{WM}}r_{j_w}^i$) planned by the GA should be later than the earliest starting time ${}^{\text{WM}}E_{r_{j_w}^i}$, which is a constraint imposed by initialization or the previous iteration. The details of the GA are explained in the next section.

The second statement ${}^{\text{DM}}o_{j_d}^i = g_1({}^{\text{WM}}r_{j_w}^i)$ denotes that once ${}^{\text{WM}}r_{j_w}^i$ has been obtained from the WM module, a procedure g_1 could be used to determine planned due dates for dyeing jobs ${}^{\text{DM}}o_{j_d}^i$. Notice that the relationship between dyeing jobs and weaving jobs is a many-to-many mapping (Fig. 3). That is, a weaving job may need more than one dyeing jobs, and a dyeing job may be used in many weaving jobs. To set up a planned due date for a dyeing job j_d^i , we first have to identify all the weaving jobs that use job j_d^i . Define $Par(j_d^i)$ as the set of weaving jobs that use dyeing job j_d^i . The function g_1 can then be described as follows:

$${}^{\text{DM}}o_{j_d}^i = g_1({}^{\text{WM}}r_{j_w}^i) = \text{Min}_{j_w^i \in Par(j_d^i)} ({}^{\text{WM}}r_{j_w}^i - I_{j_d \rightarrow w}^i)$$

Similar to the logical structure of the first statement, the third statement $({}^{\text{DM}}Z, {}^{\text{DM}}f_{j_d}^i, {}^{\text{DM}}r_{j_d}^i) = f_2({}^{\text{DM}}E_{r_{j_d}^i}, {}^{\text{DM}}o_{j_d}^i)$ denotes the input/output of the GA in the DM module. Given ${}^{\text{DM}}E_{r_{j_d}^i}$ and ${}^{\text{DM}}o_{j_d}^i$, the GA in DM module determines a near-optimal dyeing job sequence (represented by job starting time ${}^{\text{DM}}r_{j_d}^i$), and in turn computes ${}^{\text{DM}}Z$ and ${}^{\text{DM}}f_{j_d}^i$ of the job sequence. Likewise, the GA has to ensure ${}^{\text{DM}}r_{j_d}^i \geq {}^{\text{DM}}E_{r_{j_d}^i}$. This means that the starting time ${}^{\text{DM}}r_{j_d}^i$ planned by the GA should be later than the earliest starting time ${}^{\text{DM}}E_{r_{j_d}^i}$. Notice that ${}^{\text{DM}}E_{r_{j_d}^i} = 0$ for each dyeing job j_d^i in each order i because the dyeing scheduling starts at $t = 0$.

Similar to the logical structure of the second statement, the fourth statement ${}^{\text{WM}}E_{r_{j_w}^i} = g_2({}^{\text{DM}}f_{j_d}^i)$ denotes that once ${}^{\text{DM}}f_{j_d}^i$ has been obtained from the DM module, a procedure g_2 could be used to determine ${}^{\text{WM}}E_{r_{j_w}^i}$ (earliest starting time for weaving jobs in the next iteration). As stated, a weaving job may need more than one dyeing jobs. To determine ${}^{\text{WM}}E_{r_{j_w}^i}$, we first have to identify all the dyeing jobs that are input materials of weaving job j_w^i . Define $Com(j_w^i)$ as the set of dyeing jobs that are input materials of weaving job j_w^i . The function g_2 can then be described as follows:

$${}^{\text{WM}}E_{r_{j_w}^i} = g_2({}^{\text{DM}}f_{j_d}^i) = \text{Max}_{j_d^i \in Com(j_w^i)} ({}^{\text{DM}}f_{j_d}^i + I_{j_d \rightarrow w}^i)$$

For the termination check in the iteration loop, the statement $\text{Max}_{i,j_w^i} |{}^{\text{WM}}E_{r_{j_w}^i} - {}^{\text{WM}}E_{r_{j_w}^i}| \leq h$ denotes that the iterative procedure terminates whenever ${}^{\text{WM}}E_{r_{j_w}^i}$ (earliest starting time of weaving jobs—results obtained in n th iteration) is close to ${}^{\text{WM}}E_{r_{j_w}^i}$ (earliest starting time of weaving jobs—results obtained in $n+1$ th iteration). Such a termination condition may not occur. Therefore, we terminate the procedure by defining a maximum iteration number n_r .

6. Genetic algorithms

The two sub-problems in the WM and DM modules are essentially integer programming problems. Solving the two integer programs is computationally extensive. By using CPLEX

installed on a personal computer, we attempted to solve the WM module of a numerical example, described in Section 7. Results showed that the solution cannot be obtained by CPLEX after 19 h of computation. GAs are therefore developed to solve the WM and DM modules. GA, a stochastic search technique, has been widely applied to various areas including scheduling problems (Guo et al., 2006; Lee & Asllani, 2004; Ruiz & Maroto, 2006; Ruiz et al., 2005).

A GA involves four major parts: chromosome representation, fitness function, genetic operators, and selection procedure. For the two GAs in WM and DM modules, a chromosome denotes a scheduling solution. The fitness function of a chromosome denotes its performance—total tardiness of all scheduled jobs. Genetic operators are procedures designed to create new chromosomes. Selection procedure is designed to screen good or promising chromosomes.

A GA proceeds in an iterative manner. In each iteration, new chromosomes are created by applying genetic operators on a set of original chromosomes. And good or promising chromosomes are selected and taken as the original chromosomes of the next iteration. The iterative procedure continues until termination conditions are met. A typical GA iterative procedure is stated below (Gen & Cheng, 2000), where the procedure terminates if $t \geq t_{\text{max}}$ or a particular chromosome has been the best solution for over a finite number of iterations.

Procedure GA

- Step 1: Initialization. Generate P_0 , a set of N chromosomes, and set $t = 0$
- Step 2: Use genetic operators to create a set of new chromosomes, called S_t
- Step 3: Select N chromosomes out of those in $M_t = P_t \cup S_t$. Place them in P_{t+1}
- Step 4: Termination Check

If a termination condition meets, stop and output the best chromosome in P_{t+1} . Else $t = t + 1$, go to Step 2.

6.1. GA in the WM module

The GA in the WM module is to obtain ${}^{\text{WM}}r_{j_w}^i$ as stated in the Procedure Weaving_Dyeing_Scheduling. For this GA, its chromosome representation, fitness function, genetic operators, and selection procedure are described below.

To model ${}^{\text{WM}}r_{j_w}^i$, we represent a chromosome by ${}^{\text{WX}} = (\bar{x}_1, \dots, \bar{x}_k)$ where $\bar{x}_i = (x_{i1}, \dots, x_{ij}, \dots, x_{im_i})$, in which x_{ij} denotes the planned starting time of weaving job j in order i ($1 \leq i \leq k$) and order i has m_i weaving jobs. Here x_{ij} is a positive integer in an interval $[LB, UB]$, where $UB = o_i$ and $LB = {}^{\text{WM}}E_{r_{j_w}^i}$ (refer to the notation in Section 5).

Notice that a chromosome may be an infeasible solution. With ${}^{\text{WM}}r_{j_w}^i$ (the release times of weaving jobs) being available, a chromosome in fact denotes a particular schedule, which can be used to determine the capacity demand profile over time for weaving machines. If the demand profile exceeds the available tool capacity in any day, then this chromosome denotes an infeasible solution; otherwise it is a feasible one and its resulting tardiness (fitness function) can be easily obtained from the demand profile.

Two genetic operators are developed in the GA. The first one is a modified version of the crossover operator, which is used to generate two new chromosomes (called children) from two given ones (called parents). Let $A = (a_1, \dots, a_i, \dots, a_n)$ and $B = (b_1, \dots, b_i, \dots, b_n)$ be two parent chromosomes. For each child $C = (c_1, \dots, c_i, \dots, c_n)$ generated based on A and B , each gene value c_i has only two possible outcomes (either a_i or b_i) and is

randomly determined. Notice that a newly-created chromosome may be an infeasible scheduling solution. For a pair of parent chromosomes, the modified crossover operator is repeatedly performed until two children of feasible solutions are created.

The other genetic operator is a modified version of the mutation operator, which is used to generate one new chromosome from a parent chromosome. Let $A = (a_1, \dots, a_i, \dots, a_n)$ be the parent chromosome. Out of its n genes, we randomly select one, say gene i , and replace its value by two alternative methods. The first one (called r -method) is by randomly selecting a value from the feasible range of gene i . The second one (called d -method) is by deducting the gene value by 1; that is, the new value of gene i is $a_i - 1$, which implies that a weaving job is started 1 day earlier and as a result would reduce tardiness. Of the two methods, we determine which to use by probability; that is, the probability of using d -method is 0.8 and that of using r -method is 0.2. For a parent chromosome to be mutated, the modified mutation operator is repeatedly performed until one child of feasible solution is generated.

As stated, we have to select N chromosomes out of those in M_t in order to form P_{t+1} . The selection operator proceeds as follows. Firstly, we rank all the chromosomes in M_t based on their fitness functions, and place the 1st rank chromosome in P_{t+1} . Secondly, following the ranking order, we successively determine whether a chromosome should be selected. For the i th rank ($i \geq 2$) chromosome, the probability of being selected is $a(1-a)^{i-2}$, where $0 < a < 1$ is a predefined constant.

6.1.1. GA in the DM module

In the DM module, for a set of dyeing jobs (J_d) to be scheduled, we attempt to obtain a well-performed dyeing schedule in terms of total tardiness, where the tardiness of each job $j_d \in J_d$ is measured against its planned due date ($o_{j_d}^{DM}$). To do so, we have to compute $DMF_n J_{j_d}^F$ (planned completion time) for each job in a particular dyeing schedule, which in fact can be easily obtained if the corresponding sequence of the dyeing jobs has been known, as explained below.

Consider a dyeing shop with G machine types and M jobs to be scheduled. Each job can only be processed by a particular machine type, and the earliest starting time is $t = 0$. Define $m_{(g)}$ as the number of jobs processed by machine type g . For these jobs at machine type g , define a job sequence by $\pi^g = [r_{(1)}^g, \dots, r_{(m_g)}^g]$, where $r_{(j)}^g$ denotes the rank of job j in the sequence. For example, a job sequence $[r_{(1)}^g, r_{(2)}^g, r_{(3)}^g] = [3, 1, 2]$ denotes that job 2 should be firstly processed, followed by job 3, finally by job 1.

To obtain the planned completion time of each job in π^g , we proceed as follows. According to π^g , each job is successively assigned to a dyeing machine (type g), which is the earliest available and capable to dye the job. In the machine assignment, sequence-dependent setup time is considered. At the epoch of a job being assigned to a machine, we can easily determine its planned starting time as well as its planned completion time.

Therefore, in the GA of the DM module, we represent a chromosome by $\pi = [\pi^1, \dots, \pi^G]$, where $\pi^g = [r_{(1)}^g, \dots, r_{(m_g)}^g]$ denotes a job sequence at machine type $1 \leq g \leq G$. For any two different ma-

chine types (say, α and β), the performances of π^α and π^β are completely independent. That is, π^α has no effect on the performance of π^β , and vice versa.

This implies that π is composed of G segments of independent sub-chromosomes. Therefore, in the creation of a new chromosome, we generate each new sub-chromosome by independently applying the genetic operators, and finally join them together as an aggregated new chromosome in order to compute the total tardiness of the dyeing schedule.

In creating new sub-chromosomes, two types of genetic operators are used. The first one is called a *modified crossover operator*, which is designed to create two new sub-chromosomes from a pair of randomly chosen parents. The genetic operation involves two steps. Firstly, we randomly choose a crossover point, and create two new sub-chromosomes by swapping portions of the parents. For example, let $P_1 = [1, 2, 3, 4, 5]$ and $P_2 = [5, 4, 3, 2, 1]$ be the parents. Suppose the 2nd gene is the crossover point, two new sub-chromosomes $C_1 = [1, 2, 3, 2, 1]$ and $C_2 = [5, 4, 3, 4, 5]$ can be formed. Secondly, we attempt to “tune” C_1 and C_2 to make each of them a valid sequence; that is, each gene value in C_1 and C_2 can appear only once. To obtain so, we resolve the tie by giving a new order. For example, C_1 can be tuned by firstly resolving the tie between 1st and 5th gene, which may lead to a new $C_1 = [1, 2, 3, 2, 2]$. By fixing the values of 1st and 5th genes, we can further tune C_1 by resolving the tie between 2nd and 4th gene, which may result in a valid sequence $C_1 = [1, 3, 5, 4, 2]$.

The second genetic operator is called *inversion mutation operator* (Wong et al., 2000), which is designed to create a new sub-chromosome from an existing one. The operation proceeds by randomly choosing two genes and exchanging their values. For example, let $P = [4, 3, 1, 2, 5]$ be the chromosome chosen for creating a new one. Suppose 2nd and 4th genes are picked for exchange, a new sub-chromosome $C = [4, 2, 1, 3, 5]$ is then obtained.

7. Numerical example

The proposed approach is tested by scheduling examples provided by a yarn-dyed factory. The factory involves 60 weaving machines, which are functionally identical. That is, each weaving job can be processed on any weaving machine. The factory involves

Table 2 Schedules obtained in the first three iterations.

	WM	DM
1st iteration		
Tardiness (days)	17	4.83
Running time (s)	944	337
GA generations	20,000	20,000
2nd iteration		
Tardiness (days)	19	4.54
Running time (s)	168	50
GA generations	2000	2000
3rd iteration		
Tardiness (days)	17	4.54
Running time (s)	1012	50
GA generations	20,000	2000

Table 1 Information of orders.

Order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Total
Due day	8	10	10	10	10	10	15	10	12	14	30	12	15	10	5	20	
Patterns	2	2	3	7	2	8	3	1	1	4	3	4	1	1	6	3	51
Fabric rolls	2	2	3	7	2	8	10	10	10	26	6	21	1	2	6	38	154
Color yarns	5	8	5	9	8	32	7	4	2	15	6	8	3	3	7	11	133

Table 3
Tardiness in each order.

Order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Total
1st run																	
WM	0	0	0	0	0	0	0	0	0	6	0	5	0	0	0	6	17
DM	0	0	0	0	0	2.58	2.25	0	0	0	0	0	0	0	0	0	4.83
2nd run																	
WM	0	0	0	0	0	0	0	0	0	8	0	5	0	0	0	6	19
DM	0	0	0	0	0	0	3.29	0	0	1.25	0	0	0	0	0	0	4.54
3rd run																	
WM	0	0	0	0	0	0	0	0	0	6	0	5	0	0	0	6	17
DM	0	0	0	0	0	0	3.29	0	0	1.25	0	0	0	0	0	0	4.54

Table 4
Results comparison between our approach and EDD.

Scheduling algorithm	Total tardiness (days)
EDD	26
Proposed approach	17

four types of dyeing machines. Each dyeing job can only be processed by a particular type. Each type is composed of 1–3 dyeing machine and there are seven dyeing machines in total.

As shown in Table 1, a scheduling example involves 16 customer orders, which can be further deployed into 51 pattern-cloths, 154 fabric rolls (weaving jobs) and 133 color-yarn jobs (dyeing jobs). The BOM and due date of each order is also shown in the table. For example, the due date of order 1 is 8th day, and the order involves two color patterns, two fabric rolls, and five color yarns.

We use a personal computer (Pentium IV) to schedule the orders. The solution procedure as shown in Fig. 4 terminates at the third iteration. Results obtained in each of the first three iterations are shown in Table 2. From the table, readers may wonder why the results obtained in the first iteration appear to be slightly better than those obtained in the second iteration. This is due to an implicit assumption made at the first iteration—the earliest starting time for each weaving job is $t = 0$. This in fact is infeasible because any dyeing job has not been processed at this epoch. Detailed tardiness information for each customer order is shown in Table 3.

Table 2 also indicates that it takes about 42 min computation time to proceed through the three iterations. This time span is acceptable to industry according to our interviews with the yarn-dyed factory production planner. As stated, the scheduling in the WM module is an integer program. We use proprietary software CPLEX to solve the WM module, but cannot obtain a solution after taking 19 h of computation. This again confirms the merit of our GA approach.

We also justify the effectiveness of the proposed scheduling approach by the benchmark of using earliest due date (EDD) scheduling rule, which had been used in the example factory. Table 4 indicates that our approach significantly outperforms EDD, approximately 35% reduction in tardiness.

8. Conclusions

This paper presents a scheduling approach for yarn-dyed textile manufacturing. The scheduling problem is distinct in four points: multi-stage production, sequence-dependent setup times, hierarchical product structure, and group-delivery. These four scheduling features have not been considered as a whole in literature. The scheduling objective is to minimize the total tardiness of orders.

We formulate the scheduling problem as a mixed integer programming (MIP) model, which is NP-hard in complexity. To reduce

problem complexity, we decompose the scheduling problem into two sub-problems. A solution procedure for iteratively solving the two sub-problems is proposed, where a GA is developed to solve each sub-problem.

The proposed model is tested by numerical examples provided by a yarn-dyed factory. Using the proposed method to solve a typical scheduling problem in the factory takes about 42 min by using a personal computer. The proposed approach significantly outperforms the EDD method—the one that had been widely used in the yarn-dye industry.

Acknowledgement

We are grateful to Ruentex Corp. for providing this research problem and test data.

References

- Allahverdi, A., Gupta, J. N. D., & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations OMEGA. *The International Journal of Management Science*, 27, 219–239.
- Cheng, T. C. E., Gupta, J. N. D., & Wang, G. (2000). A review of flowshop scheduling research with setup times. *Production and Operations Management*, 9(3), 262–282.
- Dastidar, S. G., & Nagi, R. (2005). Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs. *Computer and Operations Research*, 32, 2987–3005.
- Ford, F. N., & Rager, J. (1995). Expert system support in the textile industry: End product production planning decision. *Expert Systems With Applications*, 9(2), 237–246.
- Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. New York: John Wiley and Sons Inc.
- Guo, Z. X., Wong, W. K., Leung, S. Y. S., Fan, J. T., & Chan, S. F. (2006). Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: A case study based on the apparel industry. *Computer and Industrial Engineering*, 50, 202–219.
- Karacapilidis, N. I., & Pappis, C. P. (1996). Production planning and control in textile industry: A case study. *Computers in Industry*, 30, 127–144.
- Lee, S. M., & Asllani, A. A. (2004). Job scheduling with dual criteria and sequence-dependent setup: Mathematical versus genetic programming. *Omega*, 32, 145–153.
- Liu, C.-Y. (1996). Scheduling flexible flow shops with sequence-dependent setup effect, decision and control. In *Proceedings of the 35th IEEE conference* (Vol. 2, pp. 1757–1762).
- Luo, X., & Chu, F. (2006). A branch and bound algorithm of the single machine schedule with sequence dependent setup times for minimizing total tardiness. *Applied Mathematics and Computation*, 183, 575–588.
- Min, L., & Cheng, W. (2006). Genetic algorithms for the optimal common due date assignment and optimal scheduling policy in parallel machine earliness/tardiness scheduling problems. *Robotics and Computer-Integrated Manufacturing*, 22, 279–287.
- Ruiz, R., & Maroto, C. (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup time and machines and machine eligibility. *European Journal of Operation Research*, 169, 781–800.
- Ruiz, R., Maroto, C., & Alcaraz, J. (2005). Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. *European Journal of Operation Research*, 165, 34–54.
- Saydam, C., & Cooper, W. D. (1995). Dye machine scheduling and roll selection. *Production and Inventory Management Journal*, 36(4), 64–70.
- Shiroma, P. J., & Niemeyer, G. (1998). Production scheduling in the textile industry: A practical approach using evolutionary algorithms with domain-dependent

- information. In *Proceedings of the 24th annual conference of the IEEE* (Vol. 1, p. 269), Industrial Electronics Society, IECON'98.
- Sun, J., & Chisman, J. A. (1991). Computer simulation of textile facilities. *Systems Engineering*, 450–452.
- Tang, Z.-B., Hammond, J. H., & Abernathy, F. H. (1994). Design and scheduling of apparel manufacturing systems with both slow and quick production lines. In *Proceedings of the 33rd conference on decision and control* (pp. 1603–1607).
- Tomastik, R. N., Luh, P. B., & Liu, G. (1996). Scheduling flexible manufacturing systems for apparel production. *IEEE Transactions on Robotics and Automation*, 12(5), 789–799.
- Wang, L., & Wang, M. (1997). A hybrid algorithm for earliness-tardiness scheduling problem with sequence dependent setup time, decision and control. In *Proceedings of the 36th IEEE conference* (Vol. 2, pp. 1219–1222).
- Wong, W. K., Chan, C. K., & Ip, W. H. (2000). Optimization of spreading and cutting sequencing model in garment manufacturing. *Computer in Industry*, 43, 1–10.
- Yang, W.-H., & Liao, C.-J. (1999). Survey of scheduling research involving setup times. *International Journal of Systems Science*, 30(2), 143–155.
- Zandieh, M., Ghomi, S. M. T. F., & Hussein, S. M. M. (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation*, 180, 111–127.