A Multiobjective Hybrid Genetic Algorithm for TFT-LCD Module Assembly Scheduling

Che-Wei Chou, Chen-Fu Chien, *Member, IEEE*, and Mitsuo Gen

***Abstract*—The thin-film transistor-liquid crystal display (TFT-LCD) module assembly production is a flexible job-shop scheduling problem that is critical to satisfy the customer demands on time. On the module assembly shop floor, each workstation has identical and non-identical parallel machines that access the jobs at various processing velocities depending on the product families. To satisfy the various jobs, the machines need to be set up as the numerous tools to conduct consecutive products. This study aims to propose a novel approach to address the TFT-LCD module assembly scheduling problem by simultaneously considering the following multiple and often conflicting objectives such as the makespan, the weighted number of tardy jobs, and the total machine setup time, subject to the constraints of product families, non-identical parallel machines, and sequence-dependent setup times. In particular, we developed a multiobjective hybrid genetic algorithm (MO-HGA) that hybridizes with the variable neighborhood descent (VND) algorithm as a local search and TOPSIS evaluation technique to derive the best compromised solution. To estimate the validity of the proposed MO-HGA, experiments based on empirical data were conducted to compare the results with conventional approaches. The results have shown the validity of this approach. This study concludes with a discussion of future research directions.**

***Note to Practitioners*—Because of short product lifecycles, cycle time reduction and on-time delivery are crucial in high-tech industries such as the TFT-LCD and semiconductor manufacturing. To address these needs in real settings, a novel multiobjective hybrid genetic algorithm (MO-HGA) was developed, hybridizing with a variable neighborhood descent (VND) algorithm as a local search and TOPSIS technique to select the best compromised solution for the TFT-LCD module assembly scheduling problem. Experiments have shown practical viability of this approach. Future studies can be done to extend the developed solution to other high-tech manufacturing industries.**

***Index Terms*—Multiobjective genetic algorithm, scheduling, thin-film transistor-liquid crystal display (TFT-LCD), TOPSIS, variable neighborhood descent.**

I. INTRODUCTION

BECAUSE OF short product lifecycles, it is crucial to rapidly respond to various customer needs and deliver products on time in high-tech industries such as the thin-film transistor-liquid crystal display (TFT-LCD) and semiconductor manufacturing industries. The TFT-LCD manufacturing is capital and technology intensive industry. Facing the fierce competitive pressures, it is important to enhance productivity and operational efficiency.

By focusing on realistic settings, a module assembly process was formulated for use in the TFT-LCD industry as a generalization of the flexible job-shop scheduling problem (FJSP). On a flexible job-shop floor, workstations employ non-identical parallel machines that exhibit distinct production velocities. An operation can be processed using an available machine from a given workstation. The TFT-LCD module assembly scheduling problem can be divided into two subproblems: the routing (i.e., assigning each operation to machines) and scheduling problems (i.e., determining the start time of each operation to machines). The following factory-specific factors complicate the TFT-LCD module assembly scheduling problem.

On the module assembly shop floor, each workstation has identical and non-identical parallel machines that access jobs at various processing velocities depending on the product family [1], [2]. The processing time of each job is based on the distinct machines, and is reflected in the various completion times.

The module assembly process is the final stage of TFT-LCD production, and then semi-finished and finished goods are shipped to the customers. During the manufacturing process of module assembling, numerous customized components are assembled onto the panel as a product. Because of the various product types, the machines need to be set up as various tools to conduct consecutive jobs among distinct product families [3], [4]. This process is typically used in TFT-LCD module assembling, in which a sequence-dependent setup time is observed [2], [5]. Consequently, increasing the numbers of customers and product families require additional machines to be set up in practice. It is crucial to provide on-time delivery and reduce cycle time in high-tech industry [6]. However, in order to satisfy the customer due date, it may increase the completion time of jobs.

This study aims to propose a multiobjective hybrid genetic algorithm (MO-HGA) to solve the TFT-LCD module assembly

Manuscript received January 14, 2014; revised March 31, 2014; accepted April 03, 2014. Date of publication May 06, 2014; date of current version June 30, 2014. This paper was recommended for publication by Associate Editor J. Li and Editor L. Shi upon evaluation of the reviewers' comments. This work was supported in part by the National Science Council, Taiwan (NSC 96-2221-E-007-048, NSC 102-2622-E-007-013, NSC 103-2622-E-007-002, NSC 102-2221-E-007-057-MY3), in part by Hsinchu Science Park (102A26), in part by the Toward World Class University Project from the Ministry of Education (102N2075E1), and in part by the Grant-in-Aid for Scientific Research (C) of the Japan Society of Promotion of Science (JSPS No. 24510219).

C.-W. Chou and C.-F. Chien are with the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: cfchien@mx.nthu.edu.tw).

M. Gen is with the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan, and so with the Fuzzy Logic Systems Institute, Fukuoka 820-0067, Japan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2014.2316193

scheduling problem, which is characterized by a sequence-dependent setup time, non-identical machines, and pursuing multiple objectives in a real manufacturing system. In addition, the present module assembly scheduling problem considers the following multiple and conflicting objectives simultaneously: minimizing the makespan, minimizing the weighted number of tardy jobs, and minimizing the total machine setup time, that are directly related to productivity and customer satisfaction. The proposed MO-HGA also hybridizes with the variable neighborhood descent (VND) as a local search and the technique for order preference by similarity to the ideal solution (TOPSIS) to derive the best compromised solution. The proposed MO-HGA was validated using several experiments based on empirical data from a leading TFT-LCD company in Taiwan. Furthermore, a multiple objective mixed integer linear programming (MO-MILP) model was developed as a benchmark to validate the proposed MO-HGA. In addition, the MO-HGA was also compared with existing metaheuristic approaches to verify its effectiveness and efficiency in an actual manufacturing environment.

The remainder of this paper is organized as follows. Section II reviews related studies contributing to the fundamentals of this approach. Section III presents a description of the TFT-LCD module assembly scheduling problem and its mathematical model. Section IV details the MO-HGA approach to solve the multiobjective TFT-LCD module scheduling problem. Section V examines the validity of the proposed MO-HGA approach with experimental results based on empirical data. Section VI concludes this study with a discussion of the findings and future research directions.

II. Literature Review

Most scheduling problems in a real-world production system involve multiple and conflicting objectives simultaneously [7]. Genetic algorithms (GAs) have been widely used to solve multiobjective scheduling problems. In particular, Pareto approach that uses the concept of domination to find the optimal Pareto results is used to address multiple objective problems. The population is ranked by the dominance rule, and a solution is assigned a fitness value based on its rank in the population [8]. Kacem *et al.* [9] proposed a Pareto approach based on the hybridization of fuzzy logic and evolutionary algorithms to solve the multiobjective FJSP problem: minimizing makespan, minimizing the maximal machine workload, and minimizing the total workload. Ho and Tay [10] presented an efficient approach which combines evolutionary algorithm and guided local search for solving the multiple-objective FJSP problem: minimizing makespan, critical machine workload, and total workload of all machines.

TFT-LCD module assembly production lines involve parallel machines on a shop floor, and non-identical parallel machines that use different process velocities; the processing time is dependent on the product family. Cochran *et al.* [11] proposed a two-stage multi-population GA that combines with dispatching rules to solve parallel machine scheduling problems that involved multiple objectives: minimizing the makespan and total weighted tardiness. Van Hop and Nagarur [12] developed a composite GA to solve the printed circuit-boards scheduling problem as non-identical parallel machines. They addressed the problem in an integrated manner by using weighted multiple objectives to manage the grouping of the boards, and the load balancing, board sequencing, and component switching for each machine.

Jobs requiring the same procedure may be due to the product family and having the same processing time in one machine. On the other hand, incompatible product families cannot be processed together using one machine. Uzsoy [13] considered a single batch processing machine scheduling problem by using incompatible job families to minimize the total completion time and makespan. Mönch *et al.* [14] proposed two decomposition approaches based on the GA and simple heuristic algorithms to minimize the total weighted tardiness on parallel batch machines involving incompatible job families and unequal job readiness times. Malve and Uzsoy [1] presented a parallel identical batch-processing machine scheduling problem based on the arrival of dynamic, incompatible job families that minimized the maximum lateness. They proposed iterative improvement heuristic methods, a GA, and related iterative heuristics embedded with the GA.

Various customers and product families require a machine setup if two consecutive jobs from distinct product families are assigned to the same machine. Allahverdi *et al.* [15] had a comprehensive review of scheduling models with setup times, and classified them into batching and non-batching, and with sequence-independent and sequence-dependent setup times. Ruiz and Maroto [16] proposed a GA to address a complex generalized flow shop scheduling problem that results from the addition of unrelated parallel machines at each stage, sequence-dependent setup times, and machine eligibility. Considering its complex problem nature of TFT-LCD and semiconductor manufacturing such as the unrelated parallel machine environment, dynamic job arrival, non-preemption, inseparable sequence-dependent setup time, multiple-resource requirements, general precedence constraint, and job recirculation, Chien and Chen [17] proposed the optimization-based schedule generator (OptSG) based on GAs and a colored timed Petri net (CTPN) for solving the generalized scheduling problems.

A number of studies have addressed TFT-LCD manufacturing scheduling problems. For example, Jeong *et al.* [7] proposed two scheduling heuristics to solve the TFT-LCD cell process scheduling problem, while minimizing the mean flow time (MF) and maximizing production progressiveness (PP). Shin and Leon [3] developed a heuristic for TFT-LCD module process scheduling problems, minimizing the total tardiness $(T_{\max})$ and the number of family setup times (S). Lee *et al.* [18] proposed a batch-based overlap-scheduling procedure based on GA involving reworking and a time buffer to enhance system performance in a TFT-LCD cell process plant, minimizing job tardiness $(T_{\max})$, machine idle time (MI), and job wait time (JW). Liu and Yih [19] developed a self-adjusted fuzzy method to solve the automated guided vehicle dispatching problem and a modified least slack time method to solve the liquid-crystal injection machine scheduling problem in the TFT-LCD cell process, by reducing the tardiness and completion time, thereby improving the efficiency of the problem. Chung *et al.* [20] proposed a mixed-integer linear programming (MILP) model to solve the aging test operation of the module

TABLE I
LITERATURES RELATE TO TFT-LCD SCHEDULING PROBLEMS

Authors	Process	Objective	Method	Operation sequence	Machine	Setup time
Jeong *et al.* [7]	Cell	*MF*, *PP*	heuristic	sequence	parallel	v
Lee *et al.* [18]	Cell	$T_{\max}$, *MI*, *JW*	GA	sequence	single	v
Liu and Yih [19]	Cell	$T_{\max}$, $C_{\max}$	heuristic	single	parallel	-
Shin and Leon [3]	Module	$T_{\max}$, S	heuristic	single	parallel	v
Chung *et al.* [20]	Module	$C_{\max}$	MILP	single	parallel	-
Pearn *et al.* [5]	Module	W_{T}	MILP	single	parallel	v
This study	Module	$C_{\max}$, W_{NT}, S_{T}	MO-HGA	sequence	parallel	v

process scheduling problem while minimizing the makespan $(C_{\max})$ under parallel machines, non-identical ready time, and arbitrary job sizes. Pearn *et al.* [5] proposed a mixed-integer linear programming model to solve the burn-in test scheduling problem in TFT-LCD Module process. They considered the constraints: an unequal ready time, batch-dependent processing times, and sequence-dependent setup times to minimize the total workload (W_{T}).

Table I lists related studies on TFT-LCD scheduling. Indeed, few studies have simultaneously addressed multiple and conflicting objectives in the complex TFT-LCD module assembly scheduling problem. Therefore, MO-HGA was developed in this study to address this scheduling problem, minimizing the makespan $(C_{\max})$, the weighted number of tardy jobs (W_{NT}), and the total machine setup time (S_{T}) subject to the constraints of incompatible product families, non-identical parallel machines, and sequence-dependent setup times.

III. MATHEMATICAL MODEL

Before further discussion, the notations used below are summarized as follows.

Indices:

i, h index of jobs, $i, h = 1, 2, \ldots, n$;

j index of machines, $j = 1, 2, \ldots, m$;

k, g index of operations, $k, g = 1, 2, \ldots, n_i$;

e,f index of product families, $e, f = 1, 2, \ldots, l$.

Parameters:

n total number of jobs;

m total number of machines;

n_i total number of operations of job i;

l total number of product families;

o_{ik} kth operation of job i;

A_{ik} set of available machines for the o_{ik};

p_{ikj} processing time of the o_{ik} on machine j;

s_{ikhgj} sequence-dependent setup time for operation o_{hg} on machine j after finished operation o_{ik};

$$a_{ikj} = \begin{cases} 1, & \text{if } o_{ik} \text{ can be accessed on machine } j \\ 0, & \text{otherwise;} \end{cases}$$

b_{ie} job i belongs to product family e;

w_i weight (priority) of job i;

d_i due date of job i;

M a large positive number.

Decision variables:

$$x_{ikj} = \begin{cases} 1, & \text{if machine } j \text{ is selected for the operation } o_{ik} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ikhgj} = \begin{cases} 1, & \text{if machine } j \text{ is needed to setup for the operation } o_{ik} \\ 0, & \text{otherwise} \end{cases}$$

$$u_i = \begin{cases} 1, & \text{if job } i \text{ is a tardy, i.e., } t^{\mathrm{c}}_{in_i} > d_i \\ 0, & \text{otherwise;} \end{cases}$$

t^{c}_{ik} completed time of o_{ik};

t^{b}_{ik} beginning time of o_{ik}.

A. TFT-LCD Module Scheduling Specifications

TFT-LCD production consists of three main stages: the array/color filter (CF) process, the cell process, and the module process. Similar to wafer fabrication, the transistors are fabricated on a glass substrate in the array/CF process with reentrant flows [2], [17], [21]. The cell process involves assembling the array and CF substrates together and filling the liquid crystal between the two substrates. The module assembly process is the final stage of TFT-LCD production, in which all customized components are assembled as finished goods. The module process stage involves five workstations in which customized components (i.e., integrated circuit (IC), printed circuit board (PCB), driver board, backlight, and chassis) are assembled onto the panel.

1) IC and PCB bonding (Workstation 1): Bonding the IC and PCB components onto the panel.
2) The 3D substrate lamination (Workstation 2): Laminating the 3D substrate on the cell panel if the product is a 3D type.
3) Semi-finished goods packing (Workstation 3): Semi-finished goods are packed to ship to the customer.
4) Component assembly and testing (Workstation 4): Assembling customized electric components onto the panel.
5) The 3D calibration (Workstation 5): Calibrating 3D product picture setting.

After the glass substrates are assembled in the cell process, the next stage in the module process typically involves bonding the IC and PCB components on the panel. The next workstation

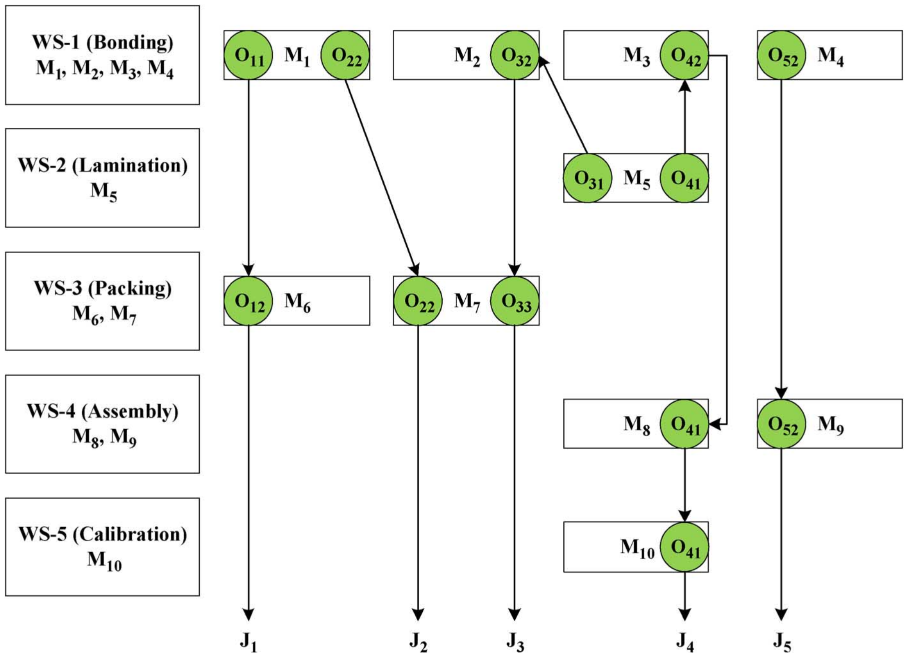

Fig. 1. Module process of the TFT-LCD manufacturing.

involves assembling other customized electric components onto the panel and conducting a test to check the reliability. Finally, the finished goods are packed and shipped.

On the module assembly shop floor, the process routes depend on the product family. Fig. 1 shows the jobs with different product families and access to different routes. For instance, for Job 4 (J_4) in Fig. 1, one of the products is a 3D-type panel, which requires laminating the 3D glass substrate onto the panel after the cell process. It must then pass through the IC and PCB bonding, components assembly, and the test workstations. Before shipping, 3D products must pass through the 3D calibration workstation to calibrate the 3D product picture settings. In addition, two types of shipments exist depending on customer demand: semi-finished and finished goods. Semi-finished goods do not require assembling the customized electric components onto the panel.

Jobs can be assigned to non-identical machines and a high-speed processing velocity is chosen to improve production efficiency. Choosing the fast processing time, the job could be finished quickly with earlier completion time. However, if all jobs are assigned to the fast processing time machine, the job must queue and also delay the completion time. Furthermore, jobs in varying product families require machine setup tools, increasing the total setup time and prolonging the job completion time.

Various product families can be produced using the varying process on a non-identical machine shop floor. The module assembly scheduling system requires selecting which job operation passes through which machine and determining the start time of each operation in each machine. Therefore, this study was conducted to solve the complex scheduling problem that considers incompatible product families, non-identical parallel machines, and sequence-dependent setup time constraints in a real production system.

B. Mathematical Modeling

The TFT-LCD module assembly scheduling problem is formulated as a multiobjective mixed-integer programming (MO-MILP) model as follows: n jobs are scheduled on m non-identical parallel machines. Each job i represents the sequence of n_i ordered operations. The kth operation of job i (o_{ik}) is assigned to machine j, that is selected from a set of available machines A_{ik}, and the operation o_{ik} accesses the processing time p_{ikj} until the operation is complete. In addition, let s_{ikhgj} be the sequence-dependent setup time between any two consecutive job h and job i that belong to distinct product families (e and f) on machine j. The module assembly scheduling problem involves allocating and assigning priorities to operations on machines, subject to the following constraints: 1) each machine can process only one operation at a time; 2) the precedence relationship of operations for each job is defined; and 3) machine setup time is required if two consecutive jobs belong to distinct product families.

Indeed, productivity and product delivery are two key performance indicators in the TFT-LCD manufacturing plant [22]. Early completion time and a low setup time can increase throughput, yielding high productivity levels. The weighted number of tardy jobs is a performance indicator to measure the on-time delivery of the products in high-tech industry such as TFT-LCD and semiconductor manufacturing [23]. Therefore, the following three criteria are considered:

1) The makespan $(C_{\max})$ of the jobs, which refers to the completion time of the latest finished job;
2) The weighted number of tardy jobs (W_{NT}): A job is considered tardy if it is not completed before its due date, and is equivalent to the percentage of on-time shipments. A low W_{NT} value is used to optimize customer satisfaction and on-time delivery;

3) The total machine setup time (S_{T}) represents the total setup time among all machines.

The mathematical programming formulation of the TFT-LCD module assembly scheduling problem is described as follows:

Objective function

$$\min C_{\max} = \max_{1 \le i \le n} \{t^{\mathrm{c}}_{in_i}\} \tag{1}$$

$$\min W_{\mathrm{NT}} = \sum_{i=1}^{n} w_i u_i \tag{2}$$

$$\min S_{\mathrm{T}} = \sum_{i=1}^{n}\sum_{k=1}^{n_i}\sum_{h=1}^{n}\sum_{g=1}^{n_h}\sum_{j=1}^{m} s_{ikhgj} y_{ikhgj} \tag{3}$$

subject to

$$C_{\max} \ge t^{\mathrm{c}}_{in_i}, \quad \forall i \tag{4}$$

$$t^{\mathrm{b}}_{ik} + p_{ikj} x_{ikj} \le t^{\mathrm{c}}_{ik}, \quad \forall i, \mathrm{k}, j \tag{5}$$

$$t^{\mathrm{c}}_{ik} \le t^{\mathrm{b}}_{i,k+1}, \quad \forall i, k = 1, 2, \ldots, n_i - 1 \tag{6}$$

$$x_{ikj} \le a_{ikj}, \quad \forall i, k, j \tag{7}$$

$$\sum_{j \in A_{ik}} x_{ikj} = 1, \quad \forall i, k \tag{8}$$

$$t^{\mathrm{c}}_{ik} + M(y_{ikhgj} - 1) \le t^{\mathrm{b}}_{hg} + M(2 - x_{ikj} - x_{hgj}), \quad \forall i, k, h, g, j, \mathrm{o}_{ik} \ne \mathrm{o}_{hg} \tag{9}$$

$$t^{\mathrm{c}}_{hg} - M y_{ikhgj} \le t^{\mathrm{b}}_{ik} + M(2 - x_{ikj} - x_{hgj}), \quad \forall i, k, h, g, j, \mathrm{o}_{ik} \ne \mathrm{o}_{hg} \tag{10}$$

$$t^{\mathrm{b}}_{ik} + p_{ikj} + s_{ikhgj} + M(\mathrm{y}_{ikhgj} - 1) \le t^{\mathrm{b}}_{hg} + M(2 - x_{ikj} - x_{hgj}), \quad \forall i, k, h, g, j, o_{ik}, \ne o_{hg} \tag{11}$$

$$t^{\mathrm{b}}_{hg} + p_{hgj} + s_{ikhgj} - M\mathrm{y}_{ikhgj} \le t^{\mathrm{b}}_{ik} + M(2 - x_{ikj} - x_{hgj}), \quad \forall i, k, h, g, j, \mathrm{o}_{ik} \ne \mathrm{o}_{hg} \tag{12}$$

$$y_{ikhgj} + y_{hgikj} = x_{ikj} x_{hgj}, \quad \forall i, k, h, g, j, \mathrm{o}_{ik} \ne \mathrm{o}_{hg} \tag{13}$$

$$t^{\mathrm{c}}_{in_i} > d_i + M(1 - u_i), \quad \forall i \tag{14}$$

$$t^{\mathrm{c}}_{in_i} \le d_i - M u_i, \quad \forall i \tag{15}$$

$$x_{ikj} \in \{0, 1\}, \quad \forall i, k, j \tag{16}$$

$$y_{ikhgj} \in \{0, 1\}, \quad \forall i, k, h, g, j, \mathrm{o}_{ik}, \ne \mathrm{o}_{hg} \tag{17}$$

$$u_i \in \{0, 1\}, \quad \forall i \tag{18}$$

$$t^{\mathrm{c}}_{ik} \ge 0, \quad \forall i, k \tag{19}$$

where

$$t^{\mathrm{b}}_{ik} = t^{\mathrm{c}}_{ik} - p_{ikj}, \quad \forall i, k, j. \tag{20}$$

Objective function (1) is used to minimize the completion time of jobs. Objective function (2) is used to minimize the weighted number of tardy jobs, optimizing customer satisfaction, and on-time delivery. Objective function (3) is used to minimize the total setup time of the machines. Constraint (4) defines makespan $C_{\max}$. Constraints (5) and (6) denote the operation precedence constraints, ensuring that each job follows the specific operation sequence. Constraints (7) and (8) guarantee that the assigned machine is selected from the set of available machines for each operation. Constraints (9) and 10 are disjunctive constraints. They show that o_{hg} cannot start before o_{ik} is completed, or that o_{hg} must be completed before the start of o_{ik} if they are assigned to the same machine j. In other words, o_{ik} cannot overlap with o_{hg}. Constraints (11) and (12) are disjunctive constraints that consider the machine setup time. The machine that has finished o_{ik} must be set up before the next incoming operation, that is, o_{hg}. Setup time (s_{ikhgj}) is required if the two consecutive jobs belong to different product families; otherwise, there is no setup time. If both operations are assigned to the same machine j, Constraint (13) forces the selection of one of the two precedent relationship of o_{ik} and o_{hg}. Constraints (14) and (15) are contingent constraints; if the completion time of job i is greater than due date d_i (i.e., $u_i = 1$), then job i is a tardy job. Otherwise, job i is shipped to the customer on time (i.e., $u_i = 0$). Constraints (16)–(18) define the binary variables of decision variables x_{ikj}, y_{ikhgj}, and u_i. Constraint (19) indicates the completion time of o_{ik} as a nonnegative real number. Constraint (20) denotes that the beginning time (t^{b}_{ik}) of o_{ik} is determined by its completion time.

```
procedure: Multi-objective Hybrid Genetic Algorithm
input: problem dataset, MO-HGA parameters
output: the best compromised schedule
begin
  t ← 0;
  generate P(t) by population initialization routine;  //P(t): population
  calculate objectives fi(P), i = 1, 2, 3 by decoding routine;
  create Pareto E(P) by non-dominated routine;  //Fast non-dominated sort method
  evaluate eval(P) by crowding distance routine;
  keep the best compromised Pareto solution by TOPSIS routine;
  while (not terminating condition) do
    create C(t) from P(t) by crossover routine;  //C(t): offspring
    create C(t) from P(t) by mutation routine;
    improve C(t) by VND routine; // improving offspring C(t)
    calculate objectives fi(P), i = 1, 2, 3 by decoding routine;
    update Pareto E(P, C) by non-dominated routine;
    evaluate eval(P) by crowding distance routine;
    select P(t+1) from P(t) and C(t) by elitist  routine & update the best
       compromised solution by TOPSIS routine;
    tune parameters by fuzzy logic controller routine;
    t ← t + 1;
  end
output the best compromised schedule
end
```

Fig. 2. Procedure of the MO-HGA.

IV. A Multiobjective Hybrid Genetic Algorithm (MO-HGA)

GAs have been widely used to solve many intractable scheduling problems [24], [25] that involve multiple and conflicting objectives. It is crucial to simultaneously consider multiple objectives to optimize the scheduling in real settings. This study developed a MO-HGA to solve the multiobjective TFT-LCD module assembly scheduling problem as illustrated in Fig. 2, which is detailed as follows.

A. Genetic Representation

A common representation of the FJSP problem is designed using a two-vector chromosome that names all the operations of a job by using the same symbol, and interprets them according to the order of occurrence in the sequence of a given chromosome [26]. The TFT-LCD module assembly scheduling problem is a combination of operation scheduling and machine assignment decisions. Therefore, in this study, the chromosome

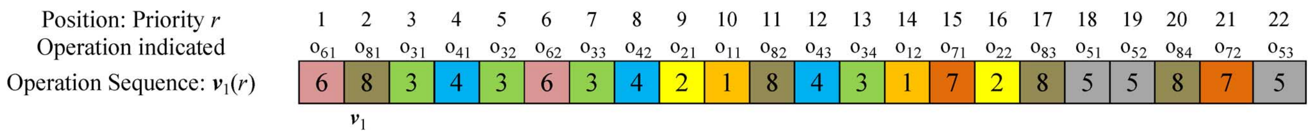

Fig. 3. Scheme of operation sequence vector $\boldsymbol{v}_1$.

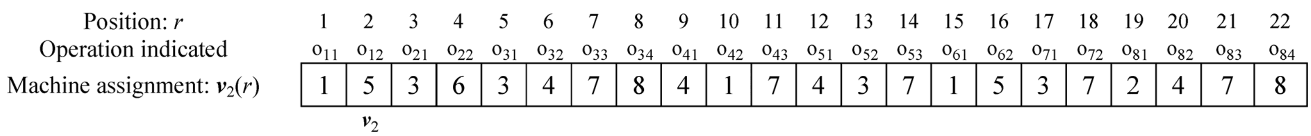

Fig. 4. Scheme of machine assignment vector $\boldsymbol{v}_2$.

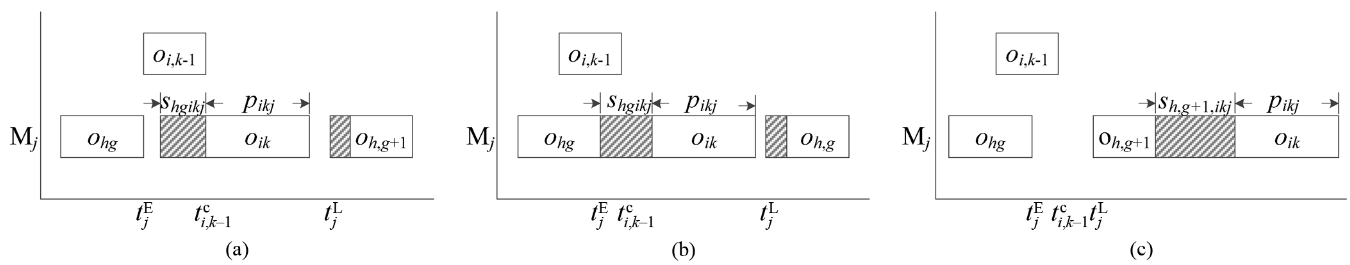

Fig. 5. Illustration of left-shift based decoding. (a) Ending of setup time earlier than $t^c_{i,k-1}$. (b) Ending of setup time later than $t^c_{i,k-1}$. (c) There is no time interval.

was designed as two parts: an operation sequence vector $(\boldsymbol{v}_1)$ and a machine assignment vector $(\boldsymbol{v}_2)$.

The evaluated TFT-LCD module assembly scheduling problem comprised eight jobs and eight machines, where each job required several operations. $\boldsymbol{v}_1$ shows that each job i appears n_i times, indicating n_i ordered operations. An example of the operation sequence vector is shown in Fig. 3. The machine assignment vector $\boldsymbol{v}_2(r)$ shows that the machine selected for the operation is indicated at position r, and shown in Fig. 4. For example, Position 1 in $\boldsymbol{v}_1(1)$ indicates o_{61} (i.e., the first operation of Job 6), and Position 1 in $\boldsymbol{v}_2(1)$ denotes that Machine 1 is assigned to o_{11}.

The main advantage of a two-vector representation is that each possible chromosome always depicts a feasible candidate.

B. Population Initialization

To guarantee the quality and diversity of an initial population, a mixed strategy is used to generate chromosomes that include an operation sequence vector and a machine assignment vector. First, a random rule strategy is applied to randomly initialize the operations in a sequence vector. Second, minimal processing time and random rule strategies are applied to generate the machine assignment vector, as follows: 1) The minimal processing time strategy [27], [28] is used to locate the machine that exhibits a minimal processing time for the permuted operation, and then adds its processing time to every subsequent entry. 2) The random rule strategy randomly assigns a machine to each operation. In this study, the random rule strategy was used to initialize the operation sequences, in which 50% of the machine assignment vectors were generated using the minimal processing time strategy, and the remaining 50% were generated using the random rule strategy.

C. Left-Shift-Based Decoding

This study used left-shift-based decoding, where each operation was shifted left until it was as compact as possible to reduce the machine idle time. This strategy is used to search for the earliest available time interval to allocate the permuted operation to a machine based on the operation sequence vector.

A time interval $[t^E_j, t^L_j]$ (i.e., beginning from t^E_j and ending at t^L_j) on machine j is available for o_{ik}, which starts as early as possible with its starting time at $\max\{t^E_j + s_{hgikj}, t^c_{i,k-1}\}$ (if $k \geq 2$) or t^E_j (if $k = 1$), where $t^c_{i,k-1}$ denotes the completion time of the previous operation. Note that $o_{i,k-1}$. o_{ik} is processed on machine j after tools or processes are set up on machine j, when the ending of the setup time is earlier than the completion time of $o_{i,k-1}$ [Fig. 5(a)] or later than $o_{i,k-1}$ [Fig. 5(b)]

$$t^b_{ik} = \begin{cases} \max\left\{t^E_j + s_{hgikj}, t^c_{i,k-1}\right\}, & \text{if } k \geq 2 \\ t^E_j, & \text{if } k = 1 \end{cases}. \tag{21}$$

Because the preceding operation o_{hg} of job h, which belongs to a different product family, was assigned to machine j, the time span must consider the setup time s_{hgikj} (i.e., job h was processed on machine j before job i). If the time span is sufficient from the beginning t^E_j to ending t^L_j, o_{ik} is allocated in the time interval $[t^E_j, t^L_j]$; otherwise, o_{ik} is allocated at the end of machine j [Fig. 5(c)], that is

$$\begin{cases} \max\left\{t^E_j + s_{hgikj}, t^c_{i,k-1}\right\} + P_{ikj} \leq t^L_j, & \text{if } k \geq 2 \\ t^E_j + P_{ikj} \leq t^L_j, & \text{if } k = 1 \end{cases}. \tag{22}$$

Left-shift-based decoding sequentially allocates each operation to an assigned machine in the order represented in the operation sequence vector.

Position: r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Operation indicated	o_{61}	o_{81}	o_{31}	o_{41}	o_{32}	o_{62}	o_{33}	o_{42}	o_{21}	o_{11}	o_{82}	o_{43}	o_{34}	o_{12}	o_{71}	o_{22}	o_{83}	o_{51}	o_{52}	o_{84}	o_{72}	o_{53}
Operation Sequence $v'_1(r)$	15	19	5	9	6	16	7	10	3	1	20	11	8	2	17	4	21	12	13	22	18	14
Machine assignment $v'_2(r)$	1	5	1	5	3	4	7	8	4	1	7	3	6	3	7	4	3	7	2	4	7	8

Fig. 6. Scheme of two-vector representation after permutation.

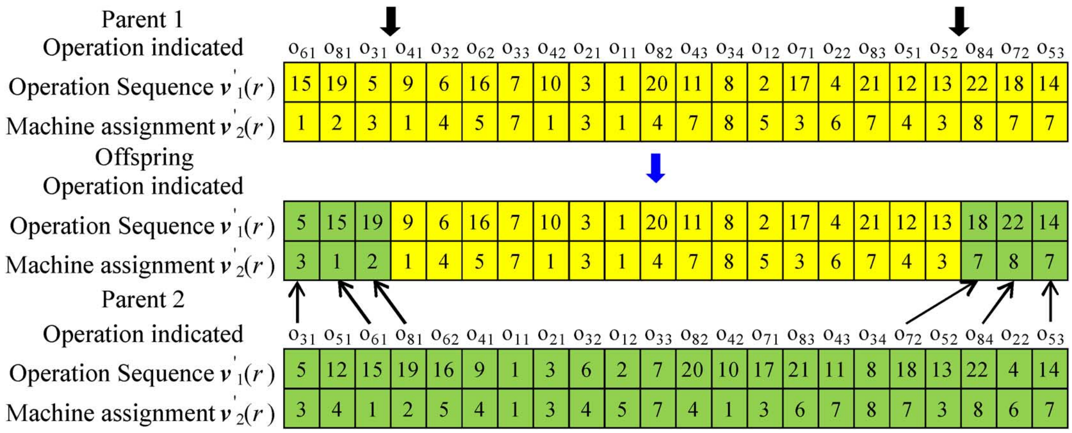

Fig. 7. Scheme of the order crossover.

To facilitate the offspring from inheriting the operation sequence from the parents, it is necessary to unify the operation sequence in the chromosome with the sequence in the decoding procedure. The operation sequence vector is reordered according to the operation starting time before the crossover and mutation procedures.

D. Crossover

In a two-vector representation, each gene in the operation sequence vector does not indicate the operation of a specific job, but refers to its context dependency. Therefore, the operation sequence vector $(\boldsymbol{v}'_1)$ is permuted to inherit their parental characteristics; the transformed two-vector representation shown in Fig. 6. $\boldsymbol{v}'_1(r)$ denotes the operation with the rth position in the permuted operation sequence vector, and $\boldsymbol{v}'_2(r)$ denotes the assigned machine for the operation (i.e., $\boldsymbol{v}'_1(1)$ is the 15th priority operation assigned to Machine 1).

Order crossover was used in this study for the operation sequence vector, and the procedure is as follows.

Step 1) Randomly select a subsection of the operation from one parent.

Step 2) Conduct an offspring by copying the subsection of the parent, including the operation sequence and machine assignment in the corresponding position.

Step 3) Delete the operations that are already in the offspring from the second parent.

Step 4) Allocate the operations and assigned machines to the unfixed positions of the offspring from left to right, according to the sequence in the second parent.

An example of this procedure is shown in Fig. 7. The offspring must transform back to its original format (two-vector representation) before it is released into the population. The permutation operation sequence vector is rearranged to $\boldsymbol{v}_1$ by replacing each operation with its job number, and the machine assignment vector is rearranged to $\boldsymbol{v}_2$.

E. Artificial Mutation

The conventional mutation operator is used to randomly generate offspring [27]. In this study, the objective functions of makespan and total machine setup time are used to minimize the production time. An artificial mutation was developed, combining the minimal processing time concept and mutation operator, reallocating the machine that exhibits the minimal processing time to the operation [28].

In the operation sequence vector, a gene of certain probability is selected and the operations are randomly exchanged with the machine that was assigned to the operation. In the machine assignment vector, the job that exhibits the longest total processing time is selected and the machine that exhibits the maximal processing time is reallocated to the minimal processing time to the corresponding operation.

The offspring generated by artificial mutation may exhibit a superior makespan compared with the makespan before the convergence was accelerated. In addition, immigration strategy was used to randomly generate new chromosomes and prohibit rapid convergence.

F. Variable Neighborhood Descent

Local search can be used to improve the convergence speed, yielding superior solutions. The VND approach is considered a local search algorithm that produces a new solution from the current population by making a slight change before it is inserted into the population [29]. The VND approach is employed to sequentially identify and exchange critical operations and find a new schedule that exhibits a small makespan in the multiobjective module assembly scheduling problem.

The makespan of a scheduling solution is defined by the length of its critical path; that is, the makespan cannot be reduced while adjusting the current critical paths. Any operation on the critical path is called a critical operation.

If o_{ik} is critical, then at least the job predecessor PJ_{ik} or the machine predecessor PM_{ik} must be critical. The preceding operation of o_{ik} is $PJ_{ik} = o_{i,k-1}$, and the successor operation of o_{ik} is $SJ_{ik} = o_{i,k+1}$. Let b_{ik}^{E} be the earliest beginning time of o_{ik} in which o_{ik} can be accessed, and the earliest completion time of o_{ik} is $c_{ik}^{\mathrm{E}} = b_{ik}^{\mathrm{E}} + p_{ijk}$, where p_{ikj} is the processing time on machine j. Let b_{ik}^{L} be the latest beginning time of o_{ik} in which o_{ik} can be processed without a delay to job completion, and the latest completion time of o_{ik} is $c_{ik}^{\mathrm{L}} = b_{ik}^{\mathrm{L}} + p_{ikj}$.

Given a time interval $[t_h^{\mathrm{E}}, t_j^{\mathrm{L}}]$, starting from t_j^{E} and ending at t_j^{L} on machine j, an assigned o_{ik} can start after the earliest completion of its job predecessor, and be completed before the latest starting time of its job successor. If o_{ik} is assigned before o_{hg} of job h on machine j, the idle time interval for the assigned o_{ik} is defined as follows:

$$\max\left\{c^{\mathrm{E}}(PM_{hg}^{j}), c^{\mathrm{E}}(PJ_{ik})\right\} + p_{ikj} + s_{ikhgj} < \min\left\{b_{hg}^{\mathrm{L}}, b^{\mathrm{L}}(SJ_{ik})\right\} \quad (23)$$

where o_{ik} can start as early as $c^{\mathrm{E}}(PM_{hg}^{j})$, and be completed as late as b_{hg}^{L} without delaying the makespan.

This study employed the VND approach to determine a schedule that yielded a small makespan. To reduce computational loading, only one critical operation is moved at a time and inserted into an available idle time interval. Therefore, the single moving operation of the VND procedure is as follows: 1) delete a critical operation r; 2) find an assignable idle time interval; and 3) allocate the deleted r into the found time interval.

G. Updating Mechanism

This study used the fast non-dominated sort routine to form the Pareto frontier. The fast non-dominated sort is one of the most efficient approaches to rank the Pareto optimality [30]. First, all solutions were evaluated according to the non-dominated routine. The first non-dominated frontier contains the dominant solutions of the population. Second, the second non-dominated frontier is iteratively constructed and the process iterates until each solution belongs to one frontier.

In a real production system, the manager must consider numerous manufacturing factors to conduct optimal scheduling. Although this study used the non-dominated routine to generate the Pareto frontier solutions, the decision maker must choose the optimal single schedule for the production system. Therefore, this study used the TOPSIS routine (please refer to the Appendix) proposed by Huang and Yoon [31] to prioritize the Pareto non-dominated solutions regarding the objectives. The schedules in the Pareto solution were ranked using the closeness value, selecting the highest value as the best compromised schedule to implement into the TFT-LCD module assembly scheduling problem.

The crowding distance routine was then used to calculate the density of solutions with the same rank to maintain the diversity of the solution set during the genetic process. The details of these calculations can be found in Deb *et al.* [30].

Finally, an elitist process was adopted to select solutions for the next generation with lower rank of the Pareto frontier. If the solutions belonged to the same frontier, then the solution with the high crowding distance located in the comparatively less crowded region was chosen, representing superior diversity. In each generation, the numbers of optimal solutions are reserved. During the elitist selection process, duplicated chromosomes are prohibited from entering the population.

TABLE II
JOB INFORMATION OF FIVE JOBS AND FIVE MACHINES

Job	Job Quantity	Product Family	Job Weighted	Operation Sequence	Due Date [k sec]
1	400	A	0.16	o_{11}, o_{12}	60.0
2	600	A	0.24	o_{21}, o_{22}	43.2
3	450	B	0.18	o_{31}, o_{32}	48.0
4	400	C	0.16	o_{41}, o_{42}, o_{43}	86.4
5	650	B	0.26	o_{51}, o_{52}	70.0

TABLE III
OPERATION PROCESSING TIME OF FIVE JOBS AND FIVE MACHINES

Processing time [k sec]	M1	M2	M3	M4	M5
o_{11}	20.0	14.0	-	-	-
o_{12}	-	-	20.0	12.0	-
o_{21}	30.0	21.0	-	-	-
o_{22}	-	-	30.0	18.0	-
o_{31}	22.5	15.8	-	-	-
o_{32}	-	-	22.5	13.5	-
o_{41}	-	-	-	-	6.0
o_{42}	20.0	14.0	-	-	-
o_{43}	-	-	20.0	12.0	-
o_{51}	32.5	22.8	-	-	-
o_{52}	-	-	32.5	19.5	-

This study used the auto-tuning strategy [32] to dynamically regulate the parameters of the MO-HGA by employing a fuzzy logic controller (FLC). Two FLCs, the crossover and mutation FLCs, were implemented to adaptively regulate the rates of crossover and mutation operators during the genetic search process. This enabled the automatic tuning of the parameters of the MO-HGA depending on the convergence situation of the current generation.

H. Numerical Illustration of MO-HGA

A small-scale case was used to demonstrate the applicability of the proposed MO-HGA. In this case, five jobs involving 11 operations were arranged to five machines on the shop floor, and the job quantity, product family, and operation sequences are listed in Table II. The operation processing time of this small-scale case is shown in Table III. The processing time of these 11 operations were designed on the basis of empirical setting that depends on the job quantity and production velocity of the machine. The sequence-dependent setup time between the product families is shown in Table IV.

The MO-HGA parameters were set as follows: a population size of 100, a maximal generation size of 1000, an initial crossover rate of 0.6, an artificial mutation rate of 0.6, and an immigration mutation rate of 0.2. Furthermore, the weight vector of the objective functions in the TOPSIS approach was $w_q = [0.2, 07, 0.1]$; this was determined by the preference of the domain expert of the objective functions. W_{NT} is the first-priority objective in real settings that is a leading TFT-LCD company in Taiwan. The MO-HGA procedure is finished when it attains the terminating condition, that is, when 30 generations

TABLE IV
SEQUENCE DEPENDENT SETUP TIME OF FIVE JOBS AND FIVE MACHINES

Product family	A		A		B		C			B	
Operation	o_{11}	o_{12}	o_{21}	o_{22}	o_{31}	o_{32}	o_{41}	o_{42}	o_{43}	o_{51}	o_{52}
o_{11}	-	-	-	-	7.2	-	-	7.2	-	7.2	-
o_{12}	-	-	-	-	-	3.6	-	-	3.6	-	3.6
o_{21}	-	-	-	-	7.2	-	-	7.2	-	7.2	-
o_{22}	-	-	-	-	-	3.6	-	-	3.6	-	3.6
o_{31}	7.2	-	7.2	-	-	-	-	7.2	-	-	-
o_{32}	-	3.6	-	3.6	-	-	-	-	3.6	-	-
o_{41}	-	-	-	-	-	-	-	-	-	-	-
o_{42}	7.2	-	7.2	-	7.2	-	-	-	-	7.2	-
o_{43}	-	3.6	-	3.6	-	3.6	-	-	-	-	3.6
o_{51}	7.2	-	7.2	-	-	-	-	7.2	-	-	-
o_{52}	-	3.6	-	3.6	-	-	-	-	3.6	-	-

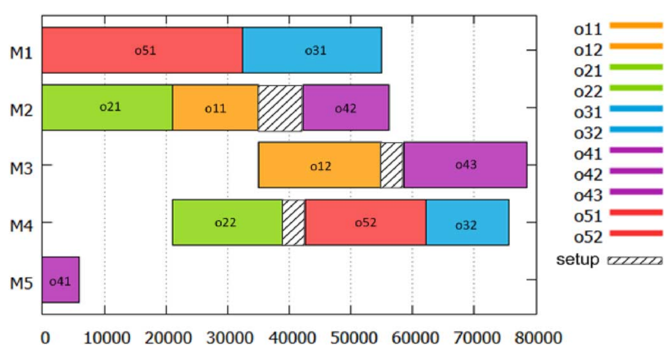

Fig. 8. Gantt chart of five jobs and five machines case.

return the same solution set. The developed MO-HGA was implemented in a C++ program, and simulated on a personal computer that exhibited the following specifications: Intel(R) Core(TM)2 Duo CPU E7500 and 4 GB RAM.

The schedule obtained using the proposed MO-HGA is described as follows: the makespan was 78.6 [k sec], the weighted number of the tardy jobs was 0.18, and the total machine setup time was 14.4 [k sec]. The Gantt chart of five jobs and five machines is shown in Fig. 8. The five jobs were associated with the product families involved in the machine setup between o_{11} and o_{42} in Machine 2, o_{12} and o_{43} in Machine 3, and o_{22} and o_{52} in Machine 4. To access the multiobjective scheduling problem in this study, the TOPSIS approach was used to evaluate the compromised schedule and implement it in the manufacturing system. In this case, four alternative schedules were contained in the Pareto solution set $(C_{\max}, W_{\mathrm{NT}}, S_{\mathrm{T}})$: (78.6, 0.18, 14.4), (87.5, 0.18, 10.8), (80, 0.24, 10.8), and (76.2, 0.26, 14.4). The TOPSIS approach considered the preference of the decision maker among the objective functions; in this case (78.6, 0.18, 14.4) was determined the best compromised solution, exhibiting the lowest weighted number of tardy jobs and ensuring that customer satisfaction is the first priority in a real manufacturing environment.

V. EXPERIMENTS

To estimate the validity of the proposed approach, for small-scale test problems, the compromised schedule was compared with the results obtained from LINGO solver. A comparison between the results of the MO-HGA and LINGO in small-scale problems shows that the MO-HGA can also be used to solve large-scale problems. For large-scale empirical problems, the proposed MO-HGA was compared with existing approaches, such as the interactive adaptive-weight genetic algorithm (iAWGA) [33] and the non-dominated sorting genetic algorithm II (NSGA-II) [30], to verify the effectiveness and efficiency of the proposed MO-HGA in realistic manufacturing settings.

A. Comparison Between MO-MILP and MO-HGA

To solve the multiobjective scheduling problem by using LINGO, a fuzzy MO-MILP (FMO-MILP) model was used to obtain the compromised solution as a benchmark, using fuzzy multiobjective programming, a fuzzy goal, and fuzzy constraints [34]–[36].

As mentioned, the weighted vector was established by the domain experts. This study attempted to introduce the same preferences in the MO-HGA and FMO-MILP. Thus, the multiobjective model was transformed into a weighted additive model. This enabled solving the FMO-MILP model by using the LINGO solver. The FMO-MILP model is as follows:

Objective function

$$\max z = w_1 z_1 + w_2 z_2 = w_3 z_3 \tag{24}$$

subject to

$$z_1 \leq \frac{C^u_{\max} - C_{\max}}{C^u_{\max} - C^*_{\max}} \tag{25}$$

$$z_2 \leq \frac{W^u_{\mathrm{NT}} - W_{\mathrm{NT}}}{W^u_{\mathrm{NT}} - W^*_{\mathrm{NT}}} \tag{26}$$

$$z_3 \leq \frac{S^u_{\mathrm{T}} - S_{\mathrm{T}}}{S^u_{\mathrm{T}} - S^*_{\mathrm{T}}} \tag{27}$$

$$w_1 + w_2 + w_3 = 1 \tag{28}$$

$$z, z_1, z_2, z_3 \in [0, 1] \tag{29}$$

$$\text{Constraints } (4) - (20)$$

where the objective function (24) was used to obtain the maximal sum z of the achievement degree of the fuzzy goal for the fuzzy decision. z_1,z_2, and z_3 denote the achievement degree of the fuzzy goal associated with the original multiobjective function. $(C^u_{\max}, W^u_{\mathrm{NT}}, S^u_{\mathrm{T}})$ is the upper tolerance limit, and $(C^*_{\max}, W^*_{\mathrm{NT}}, S^*_{\mathrm{T}})$ denotes the aspiration level for the fuzzy goals (z_1, z_2, z_3). w_1, w_2, and w_3 indicate the weighted coefficients that express the preference of the decision maker, which are [0.2, 0.7, 0.1], respectively. Constraints (25)–(29) are the linear membership functions of the fuzzy goal.

First, each objective function was individually solved and the optimal solution for each objective function was determined to build the aspiration level of this problem. Second, the upper bounds and the aspiration level for each fuzzy goal had to be calculated to establish the membership for the fuzzy goal and constraints. Finally, the FMO-MILP model was solved using the LINGO solver, and the compromised schedule was obtained in

TABLE V
SMALL-SCALE TEST PROBLEMS

Test Problem	# of Jobs	# of Machines	Total Operations	# of Product Families
P1	3	4	6	2
P2	4	4	8	3
P3	5	5	11	3

TABLE VI
EXPERIMENTAL RESULT OF FMO-MILP AND MO-HGA

Test Problem	MO-MILP (LINGO)			FMO-MILP (LINGO)	MO-HGA	Aspiration Level
	$C_{\max}$	W_{NT}	S_{T}	($C_{\max}$, W_{NT}, S_{T})	($C_{\max}$, W_{NT}, S_{T})	
P1	50.0	0.37	0	(69.7, 0.37, 7.2)	(69.7, 0.37, 7.2)	(50.0, 0.37, 0)
P2	60.5	0.34	10.8	(72.5, 0.34, 10.8)	(72.5, 0.34, 10.8)	(60.5, 0.34, 10.8)
P3	76.2	0.16	10.8	(78.6, 0.18, 14.4)	(78.6, 0.18, 14.4)	(76.2, 0.16, 10.8)

TABLE VII
OPTIMALITY GAPS OF SMALL-SCALE TEST PROBLEMS

Test Problem	FMO-MILP			MO-HGA		
	$C_{\max}$	W_{NT}	S_{T}	$C_{\max}$	W_{NT}	S_{T}
P1	39.4%	0%	-	39.4%	0%	-
P2	19.8%	0%	0%	19.8%	0%	0%
P3	3.1%	12.5%	25.0%	3.1%	11.1%	25.0%

addition to the makespan, weighted number of tardy jobs, and total machine setup time values. Three small-scale test problems in Table V were designed to benchmark the MO-MILP and MO-HGA, with the experimental results listed in Table VI.

Table VI shows a summary of the experimental results, and shows the test problem P2 (four jobs and four machines) as an example. The aspiration level (60.5, 0.34, 10.8) was obtained by solving each objective function individually. However, the aspiration level is the optimal solution for each objective goal, which cannot be achieved simultaneously. Therefore, the compromised solution solved using the FMO-MILP and the MO-HGA is (72.5, 0.34, 10.8), which achieves only the minimizing weighted number of tardy jobs and the minimizing total machine setup time, but cannot minimize the makespan simultaneously.

A quality criterion, the optimality gap, was defined to show the percentage of deviation of the values of the multiobjective programming approaches (i.e., the FMO-MILP and the MO-HGA) from the values of LINGO, according to the following equation [37]:

$$\text{Optimality Gap } (\%) = \frac{\text{Approach} - \text{LINGO}}{\text{LINGO}} \times 100. \quad (30)$$

The optimality gaps of the small-scale test problems, compared to the compromised solution of multiobjective programming approaches with an aspiration level, are shown in Table VII. The results show that the FMO-MILP and the MO-HGA yield the same optimality gaps for each test problem. These two approaches have the values close to the aspiration level.

TABLE VIII
COMPUTATIONAL CPU TIME RESULT (UNIT: SECOND)

Test Problem	MO-MILP (LINGO)			FMO-MILP (LINGO)	MO-HGA
	$C_{\max}$	W_{NT}	S_{T}	($C_{\max}$, W_{NT}, S_{T})	($C_{\max}$, W_{NT}, S_{T})
P1	29	68	1	99	19
P2	56	126	18	228	24
P3	580	740	66	1457	34

TABLE IX
LARGE-SCALE TEST PROBLEMS

Test Problem	# of Jobs	# of Machines	Total Operations	# of Product Families
P4	8	8	22	6
P5	20	8	57	15
P6	40	22	112	15
P7	60	22	157	18

The computational CPU time of the MO-MILP, FMO-MILP, and MO-HGA is shown in Table VIII. The computational time of the MO-MILP was individually solved using LINGO for each objective, and the computational time of the FMO-MILP is the sum of the solution time of the MO-MILP, because the FMO-MILP requires upper bounds and aspiration level values as parameter settings.

Despite the exponential growth of the computational CPU time caused by the FMO-MILP, the multiple objectives of the TFT-LCD problem can be solved using an acceptable CPU time when the problem is small in scale. However, in a real manufacturing system, the sizes of the scheduling problems are large and complex. Therefore, the proposed MO-HGA was validated using large, more complex test problems that simulated practical problems.

B. *Experimental Results of MO-HGA*

To verify the proposed MO-HGA in a complex manufacturing system, two medium-scale and two large-scale test problems were addressed in Table IX, using empirical data collected from a leading TFT-LCD company in Taiwan. The MO-HGA was also compared with the existing multiobjective approaches iAWGA and NSGA-II. All the test problems were run using 50 replications to eliminate any initialization bias, and the parameters of the MO-HGA were set to be the same as those of the mentioned numerical case.

In this study, the following measures were considered to evaluate the performance of the approaches [26].

- The number of obtained solutions $|S_j|$: Evaluate each solution set depending on the number of obtained solutions;
- The average compromised distance $D_{\mathrm{C}}(S_j)$ proposed in this paper: Calculate the closeness of the solutions of a Pareto set with the best compromised solution from the Pareto optimal set S^*. A small $D_{\mathrm{C}}(S_j)$ means that the compromised schedule is superior for the decision maker;
- Computational time $\mathrm{C_T}$: Evaluate the computational CPU time to obtain solutions;

where S_j is a solution set of each approach ($j = 1,2,3$), and S^* is a combination of all Pareto sets with 50 runs that were ob-

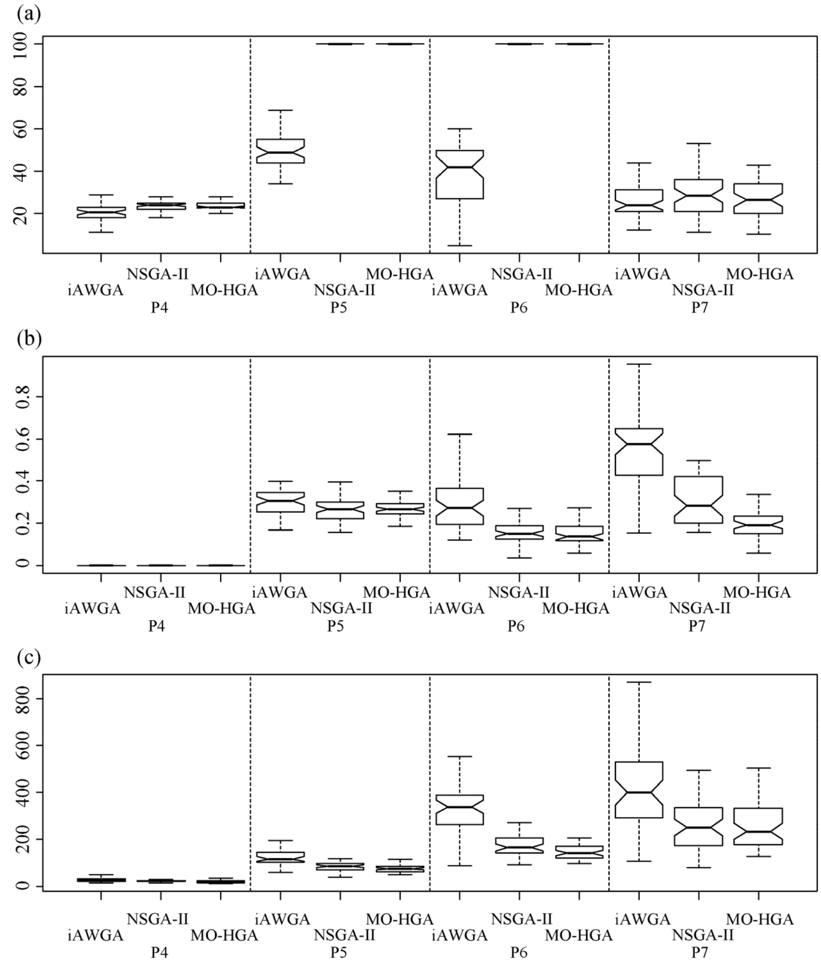

Fig. 9. $|S_j|$, $R_{\mathrm{NDS}}(S_j)$, and $D_{\mathrm{C}}(S_j)$ by iAWGA, NSGA-II, and MO-HGA. (a) Number of obtained solutions $|S_j|$. (b) Average compromised distance $D_{\mathrm{C}}(S_j)$. (c) Computational time $\mathrm{C_T}$.

TABLE X
COMPARISON RESULTS OF COMPARED APPROACHES

Test Problem	$\lvert S_j \rvert$		
	iAWGA	NSGA-II	MO-HGA
P4	20.64	23.60	23.90
P5	47.92	97.40	94.84
P6	38.60	87.84	94.46
P7	25.84	30.48	29.48

Test Problem	$D_{\mathrm{C}}(S_j)$		
	iAWGA	NSGA-II	MO-HGA
P4	0.0284	0.0070	0.0065
P5	0.3005	0.2740	0.2620
P6	0.3000	0.1734	0.1621
P7	0.5572	0.3086	0.1912

Test Problem	$\mathrm{C_T}$		
	iAWGA	NSGA-II	MO-HGA
P4	27.7	19.7	23.0
P5	132.7	80.6	83.2
P6	327.8	174.8	172.6
P7	419.5	265.8	261.2

tained using the three compared approaches. The average results of the three measures are listed in Table X, and Fig. 9 shows the

TABLE XI
SUMMARY OF KRUSKAL–WALLIS TEST FOR THREE MEASURES BY COMPARED APPROACHES

P-value	$\lvert S_j \rvert$	$D_{\mathrm{C}}(S_j)$	$\mathrm{C_T}$
P4	0.0000	0.0330	0.0000
P5	0.0000	0.0061	0.0000
P6	0.0000	0.0000	0.0000
P7	0.2962	0.0000	0.0000

box plots for $|S_j|$, $\mathrm{C_T}$, and $D_{\mathrm{C}}(S_j)$ that were obtained using the iAWGA, NSGA-II, and MO-HGA.

Regarding the statistical analysis, the Kruskal–Wallis test, which performs the equality of the population medians at $\alpha = 0.05$, offers a nonparametric alternative to one-way analysis of variance. Table XI shows the information derived from the statistical investigation of the test problems obtained using the compared approaches.

According to the computational result of $|S_j|$ in Table X and the box plots in Fig. 9, the MO-HGA and the NSGA-II can clearly address a greater number of obtain solutions than can the iAWGA for each test problem. In addition, significant differences of $|S_j|$ exist (except for P7) in the statistical analysis (Table XI).

According to the computational result of $D_{\mathrm{C}}(S_j)$ in Table X, the MO-HGA clearly dominated the other approaches for each test problem. The MO-HGA yields smaller $D_{\mathrm{C}}(S_j)$ values compared with the other approaches (i.e., the best compromised schedule obtained by the MO-HGA is suitable for the preferences of the decision maker). Furthermore, a significant difference of $D_{\mathrm{C}}(S_j)$ exists in all test problems based on the Kruskal–Wallis test result of the computational time (Table XI). This demonstrates that the MO-HGA can obtain the best compromised schedule in a real manufacturing system.

According to the computational result of $\mathrm{C_T}$ in Table X, the CPU time of the NSGA-II and MO-HGA dominated that of the iAWGA, demonstrating significantly different results in each test problem based on the statistical analysis of the computational time (Table XI). As the test problems became increasingly complex, the MO-HGA was proven faster compared with the NSGA-II.

Several non-dominated schedules are included in each test problem, and it is difficult for decision makers to choose a suitable schedule to implement in a real manufacturing system. Therefore, the proposed MO-HGA implemented TOPSIS to consider the preferences of decision makers and evaluate the compromised schedule. For instance, 28 non-dominated schedules exist in test problem P7, and the best compromised schedule is (422.5, 0.099, 392.4), that is, the compromised schedule determined the minimizing makespan as 422.5 [k sec], the minimizing weighted number of tardy jobs as 0.099, and the minimizing total machine setup time as 392.4 [k sec] in a one-week production schedule.

In practice, it is crucial to commit effective due dates of fulfilled demands to the customers. The proposed MO-HGA can assist the decision makers in selecting the compromised schedule based on the customer needs and manufacturing strategies. Moreover, big data analytics and manufacturing intelligence approaches [6], [38] can be employed for effective cycle time reduction and efficient delivery time prediction for TFT-LCD production. Also, since capacity utilization significantly affects the profitability and scheduling effectiveness of high-tech manufacturing, the developed MO-HGA can generate alternative compromised schedules that can be linked with dynamically optimized capacity expansion and migration planning in light of rolling demand forecasts [39]–[42] in which forecast errors and unfulfilled deliveries in different time periods are correlated.

VI. Conclusion

Given the fierce competition in the TFT-LCD industry, production scheduling is critical for satisfying customer demands on time to maintain competitive advantages. Focusing on real settings of TFT-LCD module assembly scheduling, this study developed the MO-HGA to obtain the best compromised solution, while considering the conflicting multiple objectives, such as the makespan, weighted number of tardy jobs, and total machine setup time.

To estimate the validity, experiments were conducted using data collected from actual settings, including medium-scale and large-scale problems, to validate the proposed MO-HGA. In addition, the MO-MILP was developed and solved using fuzzy multiobjective programming to serve as a benchmark for validating the proposed MO-HGA based on small-scale problems. Also, the MO-HGA was compared with other metaheuristic approaches to verify its effectiveness and efficiency. The proposed MO-HGA has shown its viability to generate the best compromised schedule to pursue different performance indices in a complex manufacturing system based on the preference of the decision maker.

Further studies are needed to address critical issues involved in a dynamic production system for different stages of TFT-LCD production. Future research should be done to cope with dynamic job arrivals, unpredictable machine breakdowns, and emergent job-handling situations with the tradeoffs of yield, service, and cost in a production system for fast migration of advanced technologies [43]. To manage those factory-specific factors, a rolling schedule system is required. To ensure that the schedule is followed during operation, procedures and machines that process the previous schedule may continue until completion. For example, one alternative is to set the earliest available time of jobs at the beginning of the decoding routine. Furthermore, more research should be done to employ the proposed approach to address large-scale scheduling problems in practice. The scheduling period was one week in this study, and the production planner needed to provide a one-month production schedule to prepare for work arrangements in advance. Thus, it is crucial to enhance the efficiency and the effectiveness of the proposed MO-HGA for implementation in industry in future research. In addition, in light of demand fluctuation and price erosion of TFT-LCD products, future study should be done to incorporate inventory ages, accounting principles, and bill of materials information [44] to address inventory value write-downs into production planning.

Appendix

In this study, the scales of objective functions are different dimensions in the TFT-LCD Module assembly scheduling problem, i.e., the makespan and total machine setup time are represented by seconds, and the weighted number of tardy jobs is represented by the number of jobs. The different scales will be based on the TOPSIS result. Thus, before the TOPSIS procedure, we normalize the objective value first, shown as (31)

$$f'_{pq} = \frac{f_{pq} - f_{pq}^{\min}}{f_{pq}^{\max} - f_{pq}^{\min}}, \quad \forall p \in S,\ q \in Z \tag{31}$$

where S denotes Pareto solutions, Z presents objective functions, and f_{pq} is the performance of the pth alternative in terms of the qth objective function.

The best alternative should have the shortest distance from the ideal solution (A^+) and the farthest distance from the negative-ideal solution (A^-). First of all, it needs to convert the weight vector and the objective functions evaluation vector into non-dimensional ones by the normalization (32). The weighted normalized matrix (V) is calculated by multiplying r_{pq} with the normalized weight w_q (i.e., $V = w_q r_{qp}$). The normalized weights are depended on decisions makers perform with the qth objective functions

$$r_{pq} = \frac{f'_{pq}}{\sqrt{\sum_{p=1}^{S} f'^2_{pq}}}. \tag{32}$$

The ideal solution (A^+) and negative-ideal solution (A^-) are determined with respect to minimize or maximize problems, (33), where J is associated with the positive criteria and J' is associated with the negative criteria

$$\left\{\begin{aligned} A^+ &= \left\{(\min_p v_{pq}|j \in J), (\max_p v_{pq}|j \in J')|p \in S, q \in Z)\right\} \\ &= \{v_1^+, v_2^+, v_3^+\} \\ A^- &= \left\{(\max_p v_{pq}|j \in J), (\min_p v_{pq}|j \in J)|p \in S, q \in Z\right\} \\ &= \{v_1^-, v_2^-, v_3^-\} \end{aligned}\right\}. \tag{33}$$

For each alterative, the separation distance from the ideal and negative-ideal solution are calculated in Euclidean n-dimensional distances, as (34)

$$\left\{\begin{aligned} s_p^+ &= \sqrt{\sum_{q=1}^{S} \left(v_{pq} - v_q^+\right)^2} \\ s_p^- &= \sqrt{\sum_{q=1}^{S} \left(v_{pq} - v_q^-\right)^2} \end{aligned}\right. . \tag{34}$$

Finally, the relative closeness (CL) to the ideal solution is determined by (35).

$$CL_p = \frac{S_p^-}{S_p^+ + S_p^-} \tag{35}$$

CL_p is in the interval [0, 1], and $CL_p = 1$ if and only if $A_p = A^+$, and $CL_p = 0$ if and only if $A_p = A^-$. The high closeness value, the better ranking of solution.

REFERENCES

[1] S. Malve and R. Uzsoy, "A genetic algorithm for minimizing maximum lateness on parallel identical batch processing machines with dynamic job arrivals and incompatible job families," *Comput. Oper. Res.*, vol. 34, no. 10, pp. 3016–3028, 2007.

[2] C.-F. Chien and C.-H. Chen, "A novel timetabling algorithm for a furnace process for semiconductor fabrication with constrained waiting and frequency-based setups," *OR Spectrum*, vol. 29, no. 3, pp. 391–419, 2007.

[3] H. J. Shin and V. J. Leon, "Scheduling with product family set-up times: An application in TFT LCD manufacturing," *Int. J. Prod. Res.*, vol. 42, no. 20, pp. 4235–4248, 2004.

[4] X. Li and Z. Yi, "Adaptive hybrid algorithms for the sequence-dependent setup time permutation flow shop scheduling problem," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 578–595, Jul. 2012.

[5] W. L. Pearn, J. S. Hong, and Y. T. Tai, "The burn-in test scheduling problem with batch dependent processing time and sequence dependent setup time," *Int. J. Prod. Res.*, vol. 51, no. 6, pp. 1694–1706, 2012.

[6] C. Kuo, C.-F. Chien, and J. Chen, "Manufacturing intelligence to exploit the value of production and tool data to reduce cycle time," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 103–111, Jan. 2011.

[7] B. Jeong, S.-W. Kim, and Y.-J. Lee, "An assembly scheduler for TFT LCD manufacturing," *Comput. Ind. Eng.*, vol. 41, no. 1, pp. 37–58, 2001.

[8] J. C. Tay and N. B. Ho, "Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems," *Comput. Ind. Eng.*, vol. 54, no. 3, pp. 453–473, 2008.

[9] I. Kacem, S. Hammadi, and P. Borne, "Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic," *Math. Comput. Simulation*, vol. 60, no. 3–5, pp. 245–276, 2002.

[10] N. B. Ho and J. C. Tay, "Solving multiple-objective flexible job shop problems by evolution and local search," *IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev.*, vol. 38, no. 5, pp. 674–685, Sep. 2008.

[11] J. K. Cochran, S.-M. Horng, and J. W. Fowler, "A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines," *Comput. Oper. Res.*, vol. 30, no. 7, pp. 1087–1102, 2003.

[12] N. Van Hop and N. N. Nagarur, "The scheduling problem of PCBs for multiple non-identical parallel machines," *Eur. J. Oper. Res.*, vol. 158, no. 3, pp. 577–594, 2004.

[13] R. Uzsoy, "Scheduling batch processing machines with incompatible job families," *Int. J. Prod. Res.*, vol. 33, no. 10, pp. 2685–2708, 1995.

[14] L. Mönch, H. Balasubramanian, J. W. Fowler, and M. E. Pfund, "Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times," *Comput. Oper. Res.*, vol. 32, no. 11, pp. 2731–2750, 2005.

[15] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, "A survey of scheduling problems with setup times or costs," *Eur. J. Oper. Res.*, vol. 187, no. 3, pp. 985–1032, 2008.

[16] R. Ruiz and C. Maroto, "A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility," *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 781–800, 2006.

[17] C.-F. Chien and C. Chen, "Using GA and CTPN for modeling the optimization-based schedule generator of a generic production scheduling system," *Int. J. Prod. Res.*, vol. 45, no. 8, pp. 1763–1789, 2007.

[18] S.-C. Lee, T.-S. Hsu, S.-P. Chuang, and C.-L. Yang, "Enhanced performance of overlap flow-shop scheduling involving reworking and a time buffer," *Int. J. Adv. Manuf. Technol.*, vol. 39, no. 1–2, pp. 142–150, 2008.

[19] H.-C. Liu and Y. Yih, "A fuzzy-based approach to the liquid crystal injection scheduling problem in a TFT-LCD fab," *Int. J. Prod. Res.*, vol. 51, no. 20, pp. 6163–6181, 2013.

[20] S. H. Chung, Y. T. Tai, and W. L. Pearn, "Minimizing makespan on parallel batch processing machines with non-identical ready time and arbitrary job sizes," *Int. J. Prod. Res.*, vol. 47, no. 18, pp. 5109–5128, 2009.

[21] C. Chamnanlor, K. Sethanan, C.-F. Chien, and M. Gen, "Re-entrant flow shop scheduling problem with time windows using hybrid genetic algorithm based on auto-tuning strategy," *Int. J. Prod. Res.*, vol. 52, no. 9, pp. 2612–2629, 2014.

[22] F.-M. Tseng, Y.-J. Chiu, and J.-S. Chen, "Measuring business performance in the high-tech manufacturing industry: A case study of Taiwan's large-sized TFT-LCD panel companies," *Omega*, vol. 37, no. 3, pp. 686–697, 2009.

[23] R. C. Leachman, S. Ding, and C.-F. Chien, "Economic efficiency analysis of wafer fabrication," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 4, pp. 501–512, Oct. 2007.

[24] C.-F. Chien, F. Tseng, and C.-H. Chen, "An evolutionary approach to rehabilitation patient scheduling: A case study," *Eur. J. Oper. Res.*, vol. 189, no. 3, pp. 1234–1253, 2008.

[25] J.-Z. Wu, X.-C. Hao, C.-F. Chien, and M. Gen, "A novel bi-vector encoding genetic algorithm for the simultaneous multiple resources scheduling problem," *J. Intell. Manuf.*, vol. 23, no. 6, pp. 2255–2270, 2012.

[26] M. Gen, R. Cheng, and L. Lin, *Network Models And Optimization: Multiple Objective Genetic Algorithm Approach*. London, U.K.: Springer, 2008.

[27] I. Kacem, S. Hammadi, and P. Borne, "Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems," *IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev.*, vol. 32, no. 1, pp. 1–13, Feb. 2002.

[28] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Comput. Oper. Res.*, vol. 35, no. 10, pp. 3202–3212, 2008.

[29] J. Gao, L. Sun, and M. Gen, "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems," *Comput. Oper. Res.*, vol. 35, no. 9, pp. 2892–2907, 2008.

[30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[31] C. L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*. Berlin, Germany: Springer-Verlag, 1981.

[32] Y. Yun and M. Gen, "Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics," *Fuzzy Optimization and Decision Making*, vol. 2, no. 2, pp. 161–175, 2003.

[33] L. Lin and M. Gen, "Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation," *Soft Comput.*, vol. 13, no. 2, pp. 157–168, 2009.

[34] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Manag. Sci.*, vol. 17, no. 4, pp. B-141–B-164, 1970.

[35] H. J. Zimmermann, "Fuzzy programming and linear programming with several objective functions," *Fuzzy Sets Syst.*, vol. 1, no. 1, pp. 45–55, 1978.
[36] R. N. Tiwari, S. Dharmar, and J. R. Rao, "Fuzzy goal programming—An additive model," *Fuzzy Sets Syst.*, vol. 24, no. 1, pp. 27–34, 1987.
[37] R. Z. Farahani and M. Elahipanah, "A genetic algorithm to optimize the total cost and service level for just-in-time distribution in a supply chain," *Int. J. Prod. Econ.*, vol. 111, no. 2, pp. 229–243, 2008.
[38] C.-F. Chien, C.-Y. Hsu, and C.-W. Hsiao, "Manufacturing intelligence to forecast and reduce semiconductor cycle time," *J. Intell. Manuf.*, vol. 23, no. 6, pp. 2281–2294, 2012.
[39] C.-F. Chien, C. Wu, and Y. Chiang, "Coordinated capacity migration and expansion planning for semiconductor manufacturing under demand uncertainties," *Int. J. Prod. Econ.*, vol. 135, no. 2, pp. 860–869, 2012.
[40] C.-F. Chien and J.-N. Zheng, "Mini-max regret strategy for robust capacity expansion decisions in semiconductor manufacturing," *J. Intell. Manuf.*, vol. 23, no. 6, pp. 2151–2159, 2012.
[41] C.-F. Chien and R.-T. Kuo, "Beyond make-or-buy: Cross-company short-term capacity backup in semiconductor industry ecosystem," *Flex. Service Manuf. J.*, vol. 25, no. 3, pp. 310–342, 2013.
[42] C.-Y. Lee, C.-H. Chen, and C.-F. Chien, "A simulation analysis for evaluating TFT-LCD fab capacity expansion with a distant transportation problem," *Int. J. Prod. Res.*, vol. 52, no. 6, pp. 1868–1885, 2014.
[43] C.-F. Chien, S. Dauzere-Peres, H. Ehm, J. W. Fowler, Z. Jiang, S. Krishnaswamy, T.-E. Lee, L. Moench, and R. Uzsoy, "Modeling and analysis of semiconductor manufacturing in a shrinking world: Challenges and successes," *Eur. J. Ind. Eng.*, vol. 5, no. 3, pp. 254–271, 2011.
[44] J.-Z. Wu, "Inventory write-down prediction for semiconductor manufacturing considering inventory age, accounting principle, product structure with real settings," *Comput. Ind. Eng.*, vol. 65, no. 1, pp. 128–136, 2013.

Che-Wei Chou received the B.S. and M.S. degrees in industrial engineering and management from Yuan Ze University, Taoyuan, Taiwan, in 2001 and 2003, respectively. He is working towards the Ph.D. degree at the Department of Industrial Engineering and Engineering Management, National Tsing Hua University (NTHU), Hsinchu, Taiwan.

His research interests include production planning and scheduling, modeling and analysis for manufacturing, decision analysis, and operation research.

Chen-Fu Chien (M'03) received the B.S. degree (with Phi Tao Phi Honor) with double majors in industrial engineering and electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 1990, the M.S. degree in industrial engineering and the Ph.D. degree in operations research and decision sciences from University of Wisconsin–Madison, Madison, WI, USA, in 1994 and 1996, respectively.

He was a Fulbright Scholar at the University of California–Berkeley, Berkeley, CA, USA, from 2002 to 2003 and also received PCMPCL Training at the Harvard Business School, Boston, MA, USA, in 2007. He is a Tsing Hua Chair Professor with NTHU, Taiwan. He is also the Director of the NTHU-TSMC Center for Manufacturing Excellence and the Principal Investigator for Semiconductor Technologies Empowerment Partners Consortium sponsored by National Science Council (NSC STEP Consortium), Taiwan. From 2005 to 2008, he was on-leave as the Deputy Director of Industrial Engineering Division in Taiwan Semiconductor Manufacturing Company (TSMC). He has received eight invention patents on semiconductor manufacturing and published two books, more than 120 journal papers and a number of case studies in Harvard Business School. He has been invited to give keynote speech in various conferences including APIEMS, C&IE, IEEM, IML, ISMI, KES, and leading universities worldwide. His research efforts center on decision analysis, big data analytics, modeling and analysis for semiconductor manufacturing, manufacturing strategy, and manufacturing intelligence.

Dr. Chien received the National Quality Award, the Distinguished Research Award and the Tier 1 Principal Investigator Award (Top 3%) from NSC, the Distinguished University–Industry Collaborative Research Award from the Ministry of Education, the University Industrial Contribution Awards from the Ministry of Economic Affairs, the Distinguished University–Industry Collaborative Research Award and the Distinguished Young Faculty Research Award from NTHU, the Distinguished Young Industrial Engineer Award, Best IE Paper Award, and the IE Award from the Chinese Institute of Industrial Engineering (CIIE), the Best Engineering Paper Award and Distinguished Engineering Professor by the Chinese Institute of Engineers in Taiwan, and the 2012 Best Paper Award of IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He is now an Area Editor of the *Flexible Services and Manufacturing Journal*, Editor of *Computers & Industrial Engineering*, and an Advisory Board Member of *OR Spectrum*.

Mitsuo Gen received the B.E., M.E., and the Ph.D. degrees in electronic engineering from Kogakuin University, Tokyo, Japan, in 1969, 1971, and 1975, respectively, and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 2006.

He is a Senior Research Scientist at the Fuzzy Logic Systems Institute, Iizuka, Japan, a Visiting Professor with the Department of Industrial Engineering and Engineering Management, National Tsing Hua University (NTHU), Hsinchu, Taiwan, and Professor Emeritus at the Ashikaga Institute of Technology, Ashikaga, Japan. He was a faculty member at the Ashikaga Institute of Technology, and a Professor at the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan. He was a Visiting Professor with the Department of Industrial Engineering and Operations Research, University of California, Berkeley, Berkeley, CA, USA, from 1999 to 2000, the Department of Industrial Engineering, Texas A&M University, College Station, TX, USA, in 2000, and a Hanyang Chair Professor at Hanyang University, Korea, from 2011 to 2012. His research interest includes evolutionary algorithms, manufacturing scheduling, logistics network and decision making. He is a coauthor for five books such as *Introduction to Evolutionary Algorithms* in 2010, and *Network Models and Optimization: Multiobjective Genetic Algorithm Approach* in 2008 from Springer and others. He is an Area Editor of *Computers & Industrial Engineering* and an Editorial Board member of several international journals.