

The cooperative estimation of distribution algorithm: a novel approach for semiconductor final test scheduling problems

Xin-Chang Hao · Jei-Zheng Wu · Chen-Fu Chien · Mitsuo Gen

Received: 28 July 2012 / Accepted: 16 February 2013
© Springer Science+Business Media New York 2013

Abstract A large number of studies have been conducted in the area of semiconductor final test scheduling (SFTS) problems. As a specific example of the simultaneous multiple resources scheduling problem, intelligent manufacturing planning and scheduling based on meta-heuristic methods, such as the genetic algorithm (GA), simulated annealing, and particle swarm optimization, have become common tools for finding satisfactory solutions within reasonable computational times in real settings. However, only a few studies have analyzed the effects of interdependent relations during group decision-making activities. Moreover, for complex and large problems, local constraints and objectives from each managerial entity and their contributions toward global objectives cannot be effectively represented in a single model. This paper proposes a novel cooperative estimation of dis-

tribution algorithm (CEDA) to overcome these challenges. The CEDA extends a co-evolutionary framework incorporating a divide-and-conquer strategy. Numerous experiments have been conducted, and the results confirmed that CEDA outperforms hybrid GAs for several SFTS problems.

Keywords Cooperative estimation of distribution algorithm · Manufacturing management · Flexible manufacturing systems · Semiconductor final test scheduling problems

Introduction

High-tech industries, including those of semiconductor manufacturing, LEDs, solar cells, and TFT-LCDs, are highly dynamic. In particular, the challenges in planning semiconductor manufacturing can be characterized by a high capital expenditure in capacity investment, long capacity installation lead times, high obsolescence rates caused by rapid technology development, and high demand volatility (Chien 2007; Wu 2011). Because of advanced functions with a reduced average unit cost, semiconductor applications are continuously expanding and penetrating various market segments. Smart integrated circuits (ICs) are increasingly used in medical electronics, green energy, car electronics, computers, communication, and consumer electronics. To respond to the increasing demand and challenging cost-effective supply match, the PDCCCR strategic decision-making framework is proposed to model the interrelated decision elements of semiconductor manufacturing companies, including pricing strategies (P), demand forecast and demand fulfillment planning (D), capacity planning and capacity portfolio (C), capital expenditure (C), and cost structure (C), which affect the overall financial return (R) (Chien et al. 2010a; Chien and Kuo 2012). A critical success factor is to improve capital

X.-C. Hao
Graduate School of Information, Production and Systems,
Waseda University, 2-7 Hibikino, Wakamatsu-ku,
Kitakyushu, Fukuoka 808-0135, Japan

J.-Z. Wu (✉)
Department of Business Administration, Soochow University,
56 Kueiyang Street, Section 1, Taipei 100, Taiwan, R.O.C
e-mail: jzwu@scu.edu.tw

C.-F. Chien
Department of Industrial Engineering and Engineering
Management, National Tsing Hua University, 101, Section 2,
Kuang-Fu Road, Hsinchu 30013, Taiwan, R.O.C

M. Gen
Department of Research and Development, Fuzzy Logic
Systems Institute (FLSI), 101, Section 2, Kuang-Fu Road,
Hsinchu 30013, Taiwan, R.O.C

M. Gen
Fuzzy Logic Systems Institute (FLSI), Center of Iizuka Research
and Development, 680-41 Oaza-Kawazu, Iizuka-shi,
Fukuoka 820-0067, Japan

effectiveness through the management of capital expenditures (CapEx) under uncertainty (Chien et al. 2011a).

Conventional approaches for capacity management include robust capacity strategies (Chien and Zheng 2012), new product allocation, intra-company inter- and intra-fab backup (Chien et al. 2011b), inter-company backup (Chien and Kuo 2012), outsourcing (Wu and Chien 2008a; Chien et al. 2010b; Wu et al. 2012a), collaborative design (Wu and Hsu 2009), and productivity enhancement (Chien et al. 2007a,b). Most approaches have been applied by semiconductor manufacturing companies to meet diverse and increasing demands (Shih et al. 2009). In addition to planning level enhancement, a number of efforts emphasized shop-floor-level scheduling.

Scheduling problems for semiconductor manufacturing have the features of scheduling with auxiliary resources, batching processes, multiple orders per job, internal and external scheduling of cluster tools, a large number of processing steps, random equipment failures, waiting time constraints (Chien and Chen 2007b), job shop problems with reentrance, sequence-dependent setup time (SDST), and sequence-dependent processing time (SDPT) (Uzsoy et al. 1992; Mönch et al. 2011). Flexible manufacturing systems (FMS) allow the dynamic configuration of resources to process distinct products. It is possible to assign each of these products to more than one type of unrelated resource with various efficiencies. Multipurpose resources can perform a wide range of tasks, which allows schedulers to concentrate the workloads among these resources to improve the utilization and reduce resource requirements. A possible negative effect is the reduction of production effectiveness because of the time wasted when a machine performs changeover and configuration to accommodate for the next job. Thus, it is vital to arrange a production schedule that simultaneously considers the values of multiple resources to respond to production requirements rapidly and effectively (Gerwin 1993).

The semiconductor final test scheduling (SFTS) problem is a specific type of simultaneous multiple resources scheduling (S-MRS) problems, which mainly fall into the NP-hard category (Blazewicz et al. 1983; Brucker et al. 1999). An S-MRS problem determines a process plan for each job (i.e., the determination of resource configuration) and the production schedule for multiple jobs, which informs the production team of the method, time, and sequence to effectively allocate all operations of jobs to suitable manufacturing resources (Wu and Chien 2008a,b; Li and McMahon 2007; Guo et al. 2009; Zhang and Gen 2010). Moreover, this process plan and schedule must maintain an operational feasibility and excel in all objectives considered. These challenges considerably increase the complexity of S-MRS problems. During the past several years, numerous studies have been conducted on S-MRS problems. Intelligent manufacturing planning and scheduling based on meta-heuristics, such as genetic algorithms (GAs) (Chien and Chen 2007a), simulated annealing

(SA) (Li and McMahon 2007), and particle swarm optimization (PSO) (Guo et al. 2009) have become common tools for finding satisfactory solutions within reasonable computational times in real settings. Wu and Chien (2008b) proposed a mixed-integer linear programming (MIP) model for semiconductor final testing of job scheduling and dynamic testing of machine configurations, and developed an assignment algorithm. To address the scheduling problem of multiple resources in general, Wu et al. (2012b) modeled a simultaneous multiple resource scheduling (S-MRS) problem and proposed a bi-vector encoding GA (bvGA) representing the chromosomes of operation sequence and seizing rules for resource assignment in tandem. Kim et al. (2007) developed a symbiotic evolutionary algorithm (EA) to solve integrated flexible manufacturing system scheduling problems. In their approach, the EA managed asymmetric multilevel structures, including machine allocation, tool allocation, alternative processes, and operational sequences in Level 1 (the lowest level); loading and routing plans in Level 2; loading/routing and sequencing in Level 3; and overall system integration in Level 4. This approach argued that a parallel search approach always outperforms a single sequence search.

Although previous studies have shown that the traditional EA is a powerful optimization tool that performs effectively in real applications, three primary reasons suggest that the algorithm is not entirely adequate for solving S-MRS problems. First, the population of individuals evolved in these algorithms has a strong bias to convergence in response to an increasing number of trials allocated to observed regions of the solution space with above-average fitness (Potter and de Jong 2000). Second, for S-MRS problems, a strong interdependent relationship occurs between the resource configuration (process plan) of jobs and the schedule sequence of jobs. However, interaction among decision variables of various hierarchy levels is inadequately analyzed in traditional EAs. Finally, for complex and large problems, local constraints and objectives from each managerial entity cannot be effectively represented in a single EA model.

This paper proposes a cooperative estimation of distribution algorithm (CEDA) to solve S-MRS problems. CEDA incorporates a cooperative co-evolutionary paradigm to extend the model features and improve the evaluation lead-time of EDA. In general, the overall problem is split into a set of smaller problems regarding the actual separation of management responsibilities. The decision space of a small problem is represented as subpopulations within a local evolutionary context (named species) following the population-based schema. In biological form, this concept can be considered within an ecosystem, and several species can evolve under a unique environment in a cooperative manner. Species subsequently cooperate (interact) with each other at predefined generation intervals to ensure the adaptability of the overall population within the ecosystem; that

is, the solution-searching direction aspires to both local and global objectives. For each species, the probability model is used to model the decision variables for estimating the marginal distribution of multinomial data to generate new solutions. The proposed algorithm can maintain searching diversity in the evolution, and the combination of information from the set of promising sub-solutions and the overall fitness information on the problem is used to estimate the distribution of the overall promising solutions.

The remainder of this paper is organized as follows: “Semiconductor final test scheduling problems” section provides a review of the SFTS problems; “Cooperative estimation of distribution algorithm” section presents the proposed CEDA approach in detail; “Experiments and discussion” section provides experimental comparisons that apply the CEDA approach for analyzing and solving several SFTS problems; and finally, last section offers a conclusion.

Semiconductor final test scheduling problems

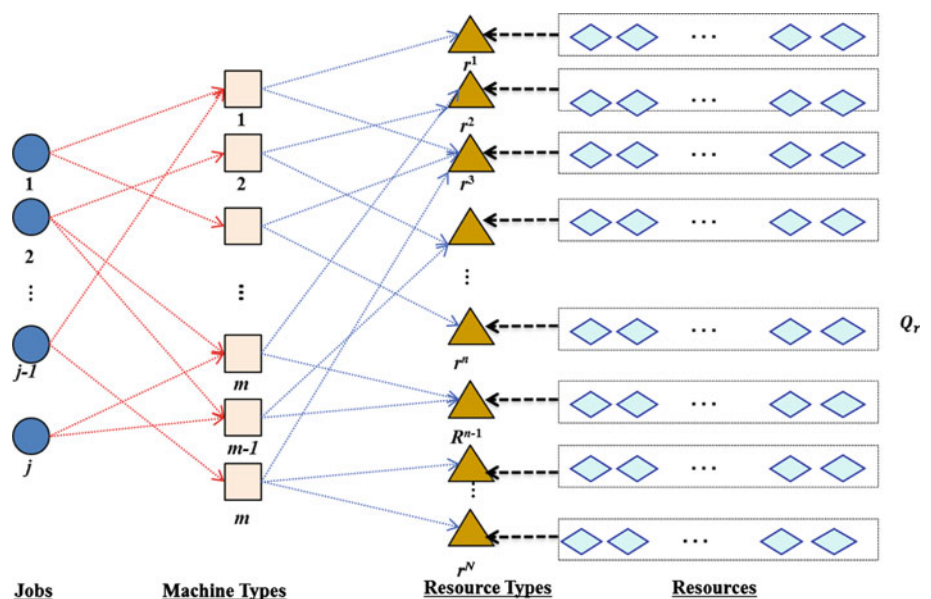
The SFTS problem shares common characteristics with S-MRS (shown in Fig. 1) and is one of the most complex and common scheduling problems (Wu and Chien 2008b). The semiconductor final test considers the IC in lots released from on-hand WIP or the forecasting output of the assembly (i.e., the preceding operation to the final test). Each job $j \in \mathbf{J}$ generally passes through the sequential processes, including a set of operations $o \in \mathbf{O}_j$ (e.g., functional test, burn-in, scan, bake, tape and reel, and package and load) for a short-term planning horizon (Uzsoy et al. 1991). Because of differing product specifications, various jobs may require differing steps of functional testing. Functional testing is the main operation site of the final test process and includes room tests, hot tests, and cold tests. After an operation starts processing a

particular resource, it must continue until completion, which is a non-preemption constraint.

Specific products can only be tested using appropriate machine configurations of testers (\mathbf{R}^1), handlers (\mathbf{R}^2), and accessories (\mathbf{R}^3). Test heads connect the tester (a central processing unit that loads test programs) to the packaged IC devices to test their functions. Handlers with built-in temperature control systems, which enable tests under various temperatures, use suitable material handling devices, called boats, to load ICs by lots packed in plastic antistatic trays or tubes. The boats release ICs from the trays or tubes, and the suction devices (called nests) lift and load the ICs onto an electrical interface (called the load board) between the packaged circuits and the tester. After the functional test, the tested ICs are placed into the classified trays (or tubes) according to the results (i.e., passed or failed). The board and the nest are the accessories (or kit). Resources for SFTS problems are expensive and can only be prepared in limited amounts; for example, $Q_r, r \in \mathbf{R}^n, n \in \{1, 2, 3\}$. Thus, the number of operations that overlap in time and require a particular resource must not exceed the total amount of available resources.

Conversely, machine configuration is dynamic. Various operations may require differing machine (configuration) types for processes, whereas more than one machine type can process one operation, which indicates the virtual unrelated parallel machine environment (Chien and Chen 2007a,b). Various machine types, $m = (r^1, \dots, r^n, \dots, r^N) \in \mathbf{M}_o, r^n \in \mathbf{R}^n, n \in \{1, 2, 3\}$ with dissimilar processing time (e.g., P_{om}) can perform an operation o . However, it is impossible to configure resources arbitrarily to form a machine type. Consequently, the universal set of all machine types is $\mathbf{M} = \bigcup_{o \in \mathbf{O}_j, j \in \mathbf{J}} \mathbf{M}_o \subseteq \mathbf{R}^1 \times \mathbf{R}^2 \times \mathbf{R}^3$. A sequence-dependent setup time (SDST), denoted by S_{lm} , is further

Fig. 1 Illustrative scheme of S-MRSs



required (Wu and Chien 2008b) to disassemble the original machine type l and assemble and calibrate the new machine type m for the incoming operation. The setup activities include resource assembly and disassembly, temperature change, software download, and calibration. The detailed modeling of SFTS problems was presented by Wu and Chien (2008b).

Cooperative estimation of distribution algorithm

In the past few decades, interest has increased in optimization methods that explicitly identify and model the optimal solution from previous generations and use the constructed model to guide further searches; these approaches are referred to as estimation of distribution algorithms (EDAs). The population-based incremental learning (PBIL) algorithm is an optimization algorithm from the EDA family (Baluja 1994). The PBIL algorithm is a combination of the EA and competitive learning. In PBIL, the real vector (probability vector) presents the probability model of decision variables instead of the individual member, and a simple incremental rule is used to update this vector after performing the selection on candidate solutions.

Most stochastic optimization algorithms, such as traditional EAs, experience dimensionality, which implies that their performance deteriorates as the dimensionality of the search space increases. Potter and de Jong (2000) first introduced the cooperative co-evolutionary paradigm to overcome this disadvantage. The basic concept of cooperative co-evolution is to divide a solution containing all decision

variables into several subcomponents. Each subcomponent represents a corresponding EA model (named species), and evolves the species sequentially (fine-grain model) or concurrently. The fitness function is evaluated by combining the representatives from other species. This approach restricts local parameters, variables, and objectives to their respective subcomponents, which considerably reduces the complexity of exploiting the search space. Plotter indicated that this decomposition achieves superior performance to basic EAs. This approach splits the solution vector into small vectors without considering the characteristics of the problem. Real-life problems often consist of multiple sub-problems divided by the actual separation of management responsibilities.

Overview of CEDA

Figure 2 shows the cooperation scheme of the proposed CEDA incorporating the co-evolutionary paradigm. First, each species d is created and initializes the probability vectors $\vec{P}_d(0)$ of its population with an equal probability value. Before sampling alternative solutions from $\vec{P}_d(0)$ in the population of species, the search context (named *partnership*), which includes the cooperator representatives of others species, must be created as a search point for each probability vector; cooperation of the species is responsible for selecting representatives from species. Subsequently, the co-evolutionary context of CEDA is initialized. Thereafter, alternative solutions $S(0)$ are sampled from the partnership and provided to the cooperation component as feedback. The cooperation component collects feedback from the species

Fig. 2 Cooperation scheme in proposed CEDA

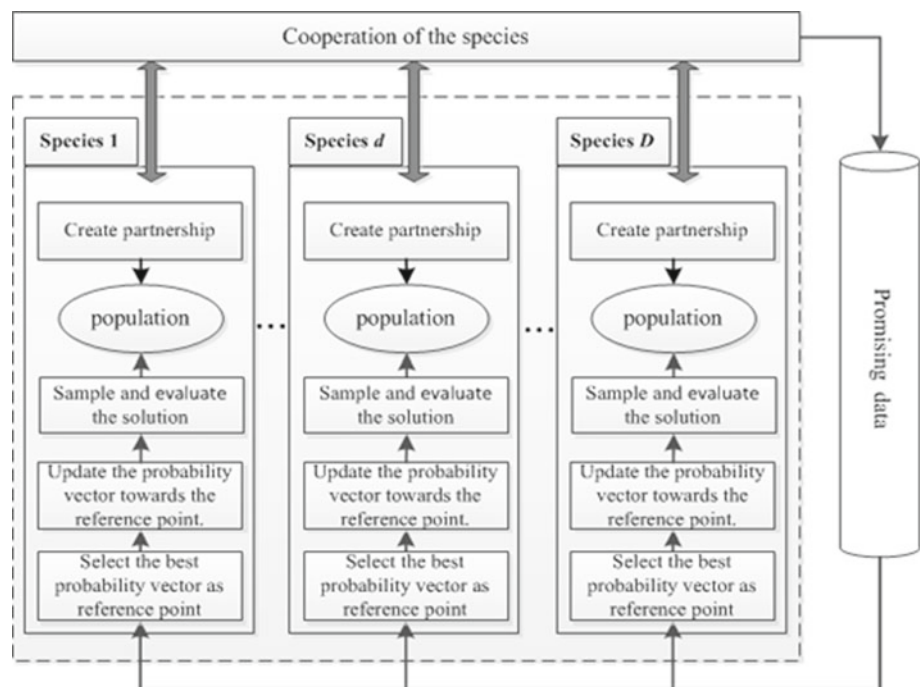


Fig. 3 Pseudo-code for CEDA

```

procedure: Cooperation Estimation of Distribution Algorithm
define:
   $g$ : number of current generations.
   $d$ : index of specie.
   $D$ : number of species.
   $pop(d)$ : population of species  $d$ .
   $prSize$ : number of the promising solutions kept by CEDA
   $elimRate$ : elimination rate of the keeping promising solutions.
   $stThreshold$ : stagnation threshold at which the partnership will be recreated.
begin
  Initialization:
     $g \leftarrow 0$ ;
    step 1: Create and initialize probability vector  $\vec{P}_d(g)$  in the population  $pop(d)$ ,  $d \in [1..D]$ .
    step 2: Create the partnership  $\vec{P}_d(g)$  for each individual in  $pop(d)$  by cooperation
      component,  $d \in [1..D]$ .
  Coevolution:
    repeat
      step 3: Sample alternative solution  $S(g)$  from the partnership
        for each probability vector in  $pop(d)$ ,  $q \in [1..D]$ .
      step 4: Cooperation component collects the alternative solution and related probability
        vectors and replace  $prSize \times elimRate$  worse items with better solutions from
         $S(g)$ .
      for each probability vector in  $pop(d)$ ,  $d \in [1..D]$  do
        step 5: Update the probability vector towards the best sample probability vector  $\vec{B}(g)$ .
        step 6: Perform mutation on probability vector to keep the diversity of sampling.
      end
      step 7: When the partnership stagnates over  $tThreshold$ , it will be recreated by
        cooperation component.
       $g \leftarrow g + 1$ ;
    until terminating criterion is met.
end;

```

and selects optimal solutions from the alternative solutions as promising data. The new probability vector $\vec{B}(0)$ is estimated using these promising data. Each species updates the probability vector toward the estimated probability vector $\vec{B}(0)$. The iteration continues until the predefined termination criteria are met. The pseudo-code for CEDA is shown in Fig. 3.

Subcomponents in CEDA

The following subsection presents the components of the proposed CEDA in detail. First, the cooperation mechanism is presented, the representation of the probability vector and learning mechanism is described, and finally, the formulation of CEDA is summarized.

Representative selection for cooperation

In CEDA architecture, each probability vector in a species only represents a partition of the overall probability vector for decision variables. The partnership is used to cooperate with other species to determine the fitness of the probability vector within the ecosystem. The representative selection from the

species is a vital function that affects the convergence and diversity of co-evolution. Tan et al. (2006) discussed two approaches. The first is the exhaustive approach; it cooperates with all individuals from other species. However, for a large-scale problem, it suffers from computational complexity. It is reasonably practicable to cooperate with only certain members of other species to estimate its cooperation extent and fitness. Tan et al. (2006) proposed another approach, in which the two optimal individuals in a species are used as the representative set of the species. However, it is often too greedy to select the optimal individual solution as representative and results in premature potentiality. For representative diversity, the k -tournament strategy (frequently $k = 2$.) was used. In contrast to the greedy strategy (optimal individual is always selected), k individuals are randomly selected from the top half of the optimal individuals in the species, and the optimal individual from the tournament with the highest fitness is selected as the representative.

Competitive learning of probability vector

For traditional EAs, the representation of a chromosome for each individual is generated by mapping the decision

space into the search space or directly encoding the decision variables; that is, the individual can be decoded to alternative solutions (Gen et al. 2008). PBEL uses the probability vector for decision variables instead of an individual member, which explicitly maintains the statistic contained in the population of an EA (Baluja and Caruana 1995). Each position in a probability vector indicates the distribution of probability regarding each variable. Generally, the domain of discrete variable X is a set of predefined values (x). In PBIL, prior knowledge of distribution is not assumed, and the distribution of random variable X has the same equal probability; the initialization is as follows:

$$P_{g=0}(X) = \frac{1}{|X|} \quad (1)$$

where $|X|$ denotes the number of values in the set of domain X .

After the initialization of CEDA, probability vectors in each species sample new alternative solutions, and the new solutions are evaluated according to a specific system objective. CEDA collects all new alternative solutions and replaces the $prSize \times elimRate$ inferior solutions in the promising data. The probability distribution of X can be estimated as follows:

$$B_g(X = x) = \frac{N(X = x) + 1/|X|}{prSize + 1/|X|} \quad (2)$$

where $N(X = x)$ denotes the number of instances in promising solutions with variable $X = x$, and $1/|X|$ represents the low bound to the probability of X .

The distribution probability of X in the probability vector is learned toward the estimated distribution of promising data, as follows:

$$P_{g+1}(X = x) = (1 - \alpha)P_g(X = x) + \alpha B_g(X = x) \quad (3)$$

where α denotes the learning rate from the current promising solutions; in particular, for $\alpha = 1$, the probability distribution is completely reconstructed by the current promising solutions.

To maintain the diversity of sampling, the distribution probability of X is updated toward the estimation distribution. The distribution can be tuned with probability p_m of the mutation which is performed using the following definition:

$$\begin{aligned} P'_g(X=x) &= \frac{P_g(X=x) + \lambda_m}{\sum_{x' \in X \setminus \{x\}} \max(P_g(x) - \lambda_m / (|X| - 1), \varepsilon) + (P_g(X=x) + \lambda_m)} \\ P'_g(X \neq x) &= \frac{\max(P_g(x) - \lambda_m / |X|, \varepsilon)}{\sum_{x' \in X \setminus \{x\}} \max(P_g(x) - \lambda_m / (|X| - 1), \varepsilon) + (P_g(X=x) + \lambda_m)} \end{aligned} \quad (4)$$

where λ_m is the mutation shift that controls the mutation operation, and ε is a small probability value to avoid the negative probability value.

Formulation of the CEDA

As shown in Fig. 3, each species in CEDA first constructs the probability vector and learns the marginal distribution probability of the variables. The learning pressure of the network depends on the size of promising solutions ($prSize$) of a species and the elimination rate of the promising solutions ($elimRate$). After constructing the probability vectors, CEDA samples the new alternative solutions to explore the promising new areas of the solution searching space. The convergence rate of CEDA is determined by the number of samplings and prior individuals in the species. For a larger number of new individual samplings, the parameter of the probability vector is biased to change frequently and increases the chance of premature convergence because of the loss of population diversity of the search space to be explored. Conversely, the parameters of the probability vector are reconstructed when CEDA moves to the next evolutionary iterator; a trade-off between the prior parameters and new parameters must be considered, and is represented by the coefficient parameter α .

Experiments and discussion

Application of CEDA for S-MRSs

Illustration of S-MRS division

S-MRS problems can be defined as the following decision-making process: *operation sequences*: determine the executing sequence of all operations required for the jobs to ensure that the precedent relationships among all operations are not violated; and *machine setup planning*: determine the accessory resources setup planning for operation according to the feature geometry and the available machining resources. Thus, a solution can be represented by the processing sequence of operations on the machines and the machine assignment of operations on the machines.

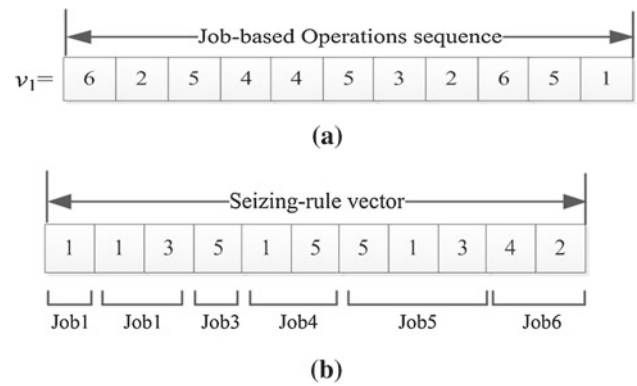
Following the schema of CEDA, S-MRSs can be mapped into two species. The first species (sequence species) exploits the scheduling problem, including all jobs. The representation of the sequence species uses job-based encoding (Gen and Cheng 2000), and the length of the chromosome equals the total number of operations. The job number denotes the operation of each job, and the k th occurrence of a job number refers to the k th operation in the sequence of this job. The illustrative representation of an SFTSP example (Table 1) is shown in Fig. 4a. For a job-based operations sequence vector $v_1 = [6, 2, 5, 4, 4, 5, 3, 2, 6, 5, 1]$, the operations sequence can be interrupted as follows: (6,1), (2,1), (5,1), (4,1), (4,2), (5,2), (3,1), (2,2), (6,2), (5,3), (1,1).

Table 1 Illustrative job and operation information for semiconductor final testing scheduling problems

Job	Operation	Machine	Processing time
1	(1,1)	4	4
2	(2,1)	2	2
		4	4
	(2,2)	2	1
3	(3,1)	4	1
4	(4,1)	1	4
	(4,2)	1	2
		4	6
5	(5,1)	4	5
	(5,2)	2	4
		4	5
	(5,3)	4	5
6	(6,1)	1	2
	(6,2)	3	5

The other species is the machine configuration species, which exploits the machine assignment of the related jobs by using seizing rules. The machine configuration species exploit the probability of seizing rules of the operations. For S-MRS, one machine required multiple resource types, each containing homogeneous, yet unrelated resources. Hence, to evaluate a candidate machine configuration, it is crucial to examine the statuses of specific resource items under desired resources. Resource items can subsequently be seized to configure a machine simultaneously. The concept was further expanded in a procedure in which the earliest available times and current machine configurations were essential statuses of resource items and were recorded and updated through computation. For example, the RND rule chose a random machine configuration to process an operation, whereas the SPT rule chose a machine configuration with the shortest processing time. It was reasonable to maintain the processing operations when the resources were available. Therefore, required resource items with the earliest available times were ready to be seized, whereas the available time of the machine was the maximal time among the earliest available times. Similarly, when applying the ECT, SST, or SSPT rule, SDST must include current machine configurations of resource items under desired resources. When multiple items are available for seizing, choosing the item with the earliest available time close to the available time of the chosen machine can break the tie to avoid early idle occupancies of resource items. Choosing the resource item requiring the minimal SDST can also break the tie (Wu et al. 2012b).

For symbolic representation of machine configuration species, set seizing rules $(1, 2, 3, 4, 5) = (\text{RND}, \text{ECT}, \text{SPT}, \text{SST}, \text{SSPT})$.

**Fig. 4** Illustration of the representation of a feasible solution for SMRSP. **a** Representation of sequence species, **b** representation of machine configuration species

SST, SSPT). The representation of machine configuration species uses seizing rules for all of the operations, and the length of the chromosome equals the total number of operations. An illustrative representation of an SFTSP example (Table 1) is shown in Fig. 4b.

Illustration of probability distribution model

Compared to traditional EAs, the traditional EDA generates a new alternative solution according to the probability model. Therefore, the probability model has a considerable effect on the performances of CEDA. Two probability models were used to estimate the probability distribution; that is, the operation sequence importance matrix and seizing-rule matrix.

The element $p_{jk}^s(g)$ of the operation sequence importance matrix represents the probability that job j will be scheduled at the k th locus of the operations sequence vector at generation g . At the beginning of the evolution of CEDA, p_{jk}^s is initialized as $p_{jk}^s(0) = 1/|J|$ for all job $j \in J$ and $k \in \{1 \dots |J_o|\}$, where J is set of jobs, and J_o is set of all the operations related to J . The sample shown in Table 1 has six jobs and 11 operation sequences. Therefore, the value of element p_{jk}^s is initialized with $1/6$ for $k \in \{1 \dots 11\}$. The element $p_{oj\rho}^A(g)$ of the seizing-rule matrix represents the probability that the operation o_j of job j will be assigned to the machine, decided by seizing rule ρ at generation g . At the beginning of the evolution of CEDA, $p_{oj\rho}^A$ is initialized as $p_{oj\rho}^A(0) = 1/|\text{Rules}|$ for all jobs $j \in J$ and $o_j \in O_j$, where J is a set of jobs, and O_j is set of all the operations of job j . Considering the sample, the value of element $p_{oj\rho}^A$ is initialized with $1/5$ for $\rho \in \{\text{RND}, \text{ECT}, \text{SPT}, \text{SST}, \text{SSPT}\}$.

Illustration of initialization of CEDA

Before the beginning of the evolution of CEDA, individuals in sequence species and machine configuration species only

Table 2 The technical specification for test problems

Problem	# of jobs	# of time	Processing time	Setup time	# of configuration	# of resource types	# of resources
Simple-problem (S)	15	1	{1,..., 6}	{1,2}	{1,2}	R^1, R^2	$R^1 : \{1, 2, 3\},$ $R^2 : \{1, 2, 3\}$
Large-scale (LS)	100	3	{1,..., 15}	{1,..,5}	{1,2,3}	R^1, R^2, R^3	$R^1 : \{10, 5, 3\},$ $R^2 : \{10, 8, 4\},$ $R^3 : \{7, 7, 5, 5\}$
Wide-range (WR)	60	3	{1,..., 50}	{1,..,15}	{1,2,3}	R^1, R^2, R^3	$R^1 : \{10, 5, 3\},$ $R^2 : \{10, 8, 4\},$ $R^3 : \{7, 7, 5, 5\}$

R^1 tester, R^2 handler, R^3 accessory

represent a partition of the overall candidate solution. For each individual in a species, a partnership must be created to determine the fitness of the individual within the ecosystem. The two-tournament representative selection was used to select an individual as the partner for cooperation. Each individual in a species can be evaluated within the partnership, and the new alternative solution is generated. For an individual that has a chromosome represented as shown in Fig. 4a, the chromosome of its partner from the machine configuration species is represented as shown in Fig. 4b. The alternative solution can be generated by following the seizing-rule decoding procedure proposed by Wu et al. (2012b).

Experimental setup

To examine the practical viability and efficiency of the proposed CEDA, we designed a numerical study to compare CEDA with efficient algorithms from previous studies. The proposed CEDA was compared with wcGA (Wu and Chien 2008b) and bvGA (Wu et al. 2012b) based on a set of simulation data of semiconductor final testing scheduling problems. The data are available at the Decision Analysis Lab (http://dalab.ie.nthu.edu.tw/newsen_content.php?id=0) website. Parameters of simulation data were aligned to provide a comparative basis. The technical specifications are shown in Table 2. The experiments were conducted on a personal computer with an Intel Core I5 CPU at 2.8 GHz and 2 GB RAM. The parameters and strategies of related algorithms are categorized in Table 3.

Results and discussion

Performance

The five larger-scale problems and five wide-range problems were used to evaluate CEDA performance compared with current meta-heuristics, wcGA, and bvGA. Table 4 shows that CEDA algorithms exhibited superior performance

to wcGA and bvGA in all experiments. The vbGA proposed the seizing-rules heuristic method, which considers the sequence-dependent changeover between machines; a previous study showed that bvGA exhibited superior performance to the traditional wcGA meta-heuristic. In contrast to the evolutionary operator of wcGA and bvGA, CEDA uses the estimated probability distributions of decision variables relating to operation sequence and machine setup planning. It provides a prediction mechanism on the variant of decision variables. Analysis of variance (ANOVA) demonstrated that the performance differences between bvGA and CEDA were significant ($P < .05$), although problem instantiations contributed more to variation (Tables 5, 6). Figures 5 and 6 shows box plots for the experimental results of the LS and WR problems. As shown in the figures, CEDA achieved superior stability to random strategy-based algorithms, although the accuracy of prediction was affected by the promising solutions.

Computation cost

The computational costs of evolutionary-based algorithms mainly depend on the number of fitness evaluations. The difference of time complexity among wsGA, bvGA, and the proposed approach mainly relies on the operators. CEDA samples a new candidate solution and improved the current candidate solution according to probability distribution. Moreover, it estimates the univariate margin distribution of decision variables using the promising data. The computational cost for training the probability model can be roughly estimated using the following equation:

$$\text{COST} \propto |\text{num of gen}| \times |\text{num of popSize}| \\ \times |\text{learning frequency}| \times |\text{dimensionality of decision variables}|$$

The average CPU times of bvGA on the LS and WR problems were 175.86 and 94.93 s, respectively. The computational times of the wcGA for LS and WR problems were

Table 3 Parameters and strategies of wcGA, bvGA and CEDA

	wcGA (Wu and Chien 2008b)	bvGA (Wu et al. 2012b)	CEDA
Iteration	4,000	4,000	4,000
Population	200	200	50
Selection	Tournament (k)	Tournament (k)	–
Strategy	Elitism	Elitism	–
Operators parameters	Crossover (P_m)	Crossover (P_m)	Sampling
	Mutation (P_c)	Mutation (P_c)	Mutation (P_m)
	$P_m = 0.30$	$P_m = 0.30$	$prSize = 100$
	$P_c = 0.65$	$P_c = 0.65$	$stThresHold = 20$
	$k = 2$	$P = 0.10$	$\alpha = 0.5$
		$k = 2$	$P_m = 0.4 \ k = 2$

Table 4 Comparison of wcGA, bvGA and CEDA on Makespan

Problems	wcGA		bvGA		CEDA	
	Mean	SD	Mean	SD	Mean	SD
LSI	127.33	1.42	124.20	1.72	121.83	1.39
LS2	143.30	2.07	137.30	2.22	135.20	1.76
LS3	135.27	3.57	131.13	1.94	123.60	1.99
LS4	158.80	2.15	148.63	2.41	137.87	2.16
LS5	143.97	2.02	142.47	1.84	138.23	1.67
WR1	322.70	5.45	307.20	4.74	302.63	4.05
WR2	248.37	5.65	240.53	3.69	234.83	3.47
WR3	280.70	4.05	275.40	3.93	269.03	3.99
WR4	229.23	4.57	220.20	3.97	209.30	4.56
WR5	284.37	6.46	260.17	4.01	246.63	4.01

Table 5 ANOVA table for the large-scale problem

Source of variation	SS	df	MS	F	P value
Sample	18,415.74	4	4,603.93	946.86	4.72E–165
Methods	2,415.85	1	2,415.85	496.85	8.17E–65
Interaction	209.13	4	52.28	10.75	3.86E–08
Error	1,489.39	149	10.00		
Total	22,450.79	299			

Table 6 ANOVA table for the wide-range problem

Source of variation	SS	df	MS	F	P value
Samples	431,522.89	4	107,880.72	5,186.36	1.76E–561
Methods	31,857.60	2	15,928.80	765.78	3.09E–143
Interaction	7,862.01	8	982.75	47.25	1.53E–54
Error	7,467.56	377	19.81		
Total	480,290.86	449			

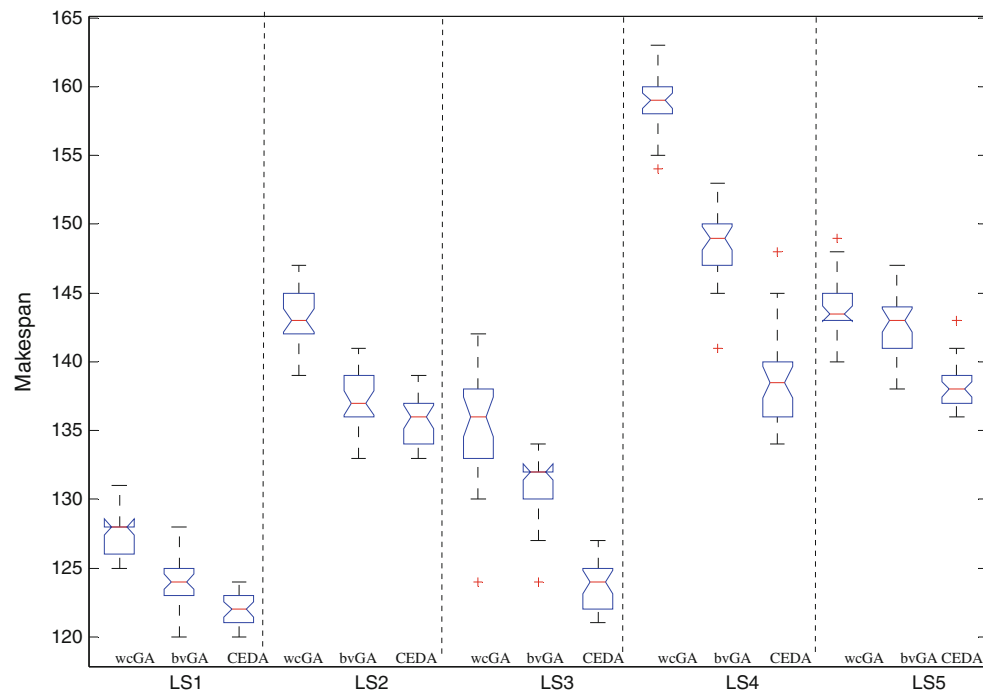


Fig. 5 Box plot for large-scale experiments

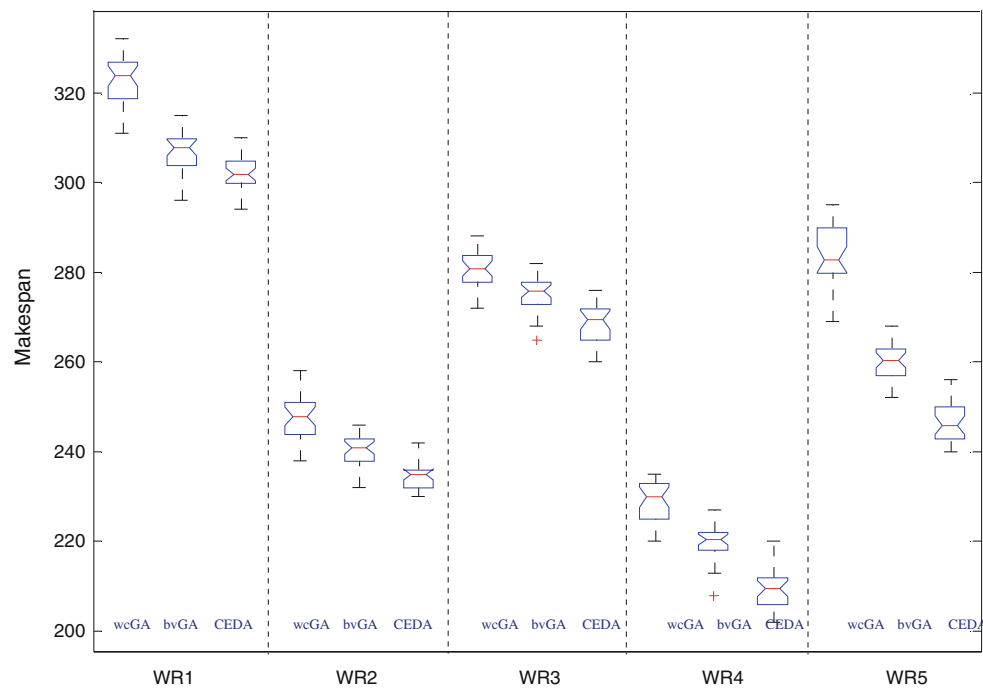


Fig. 6 Box plot for wide-range experiments

448.83 and 179.82 s, respectively. Because CEDA runs as a fine-grain scheme (species run sequentially), the computational times of CEDA for LS and WR problems were 194.85 and 105.87 s, respectively. The results show that CEDA achieved similar computational efficiency to bvGA.

Parameter tuning

The parameters of CEDA are roughly categorized into three groups: estimating distribution of the variables, sampling operator, and cooperation control. In CEDA, it is crucial to

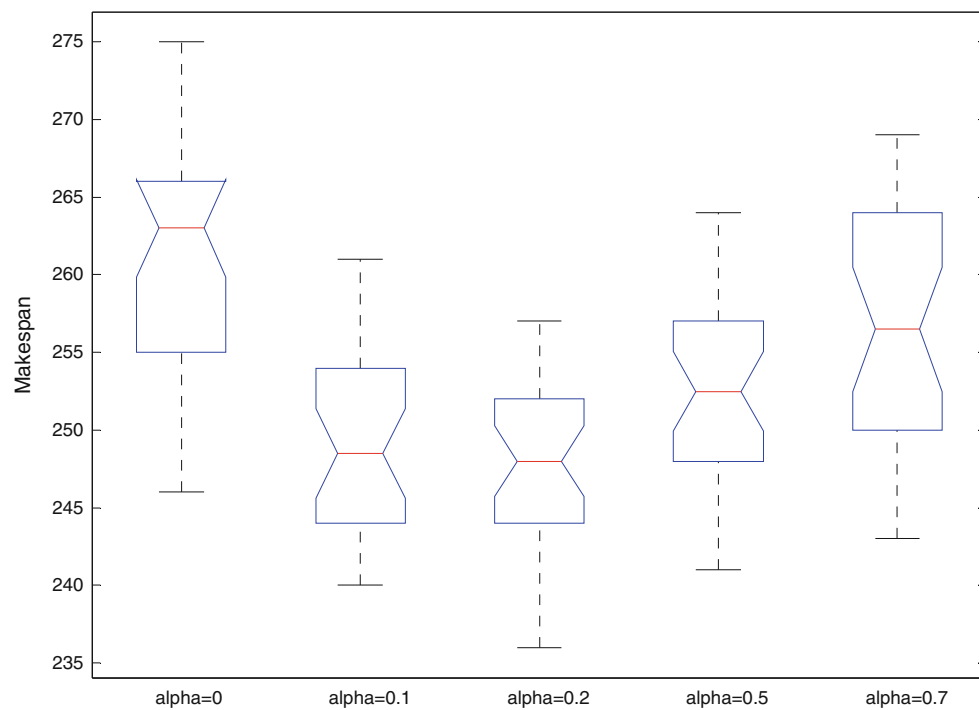


Fig. 7 Boxplot of Makespan by different learning ratings (WR5)

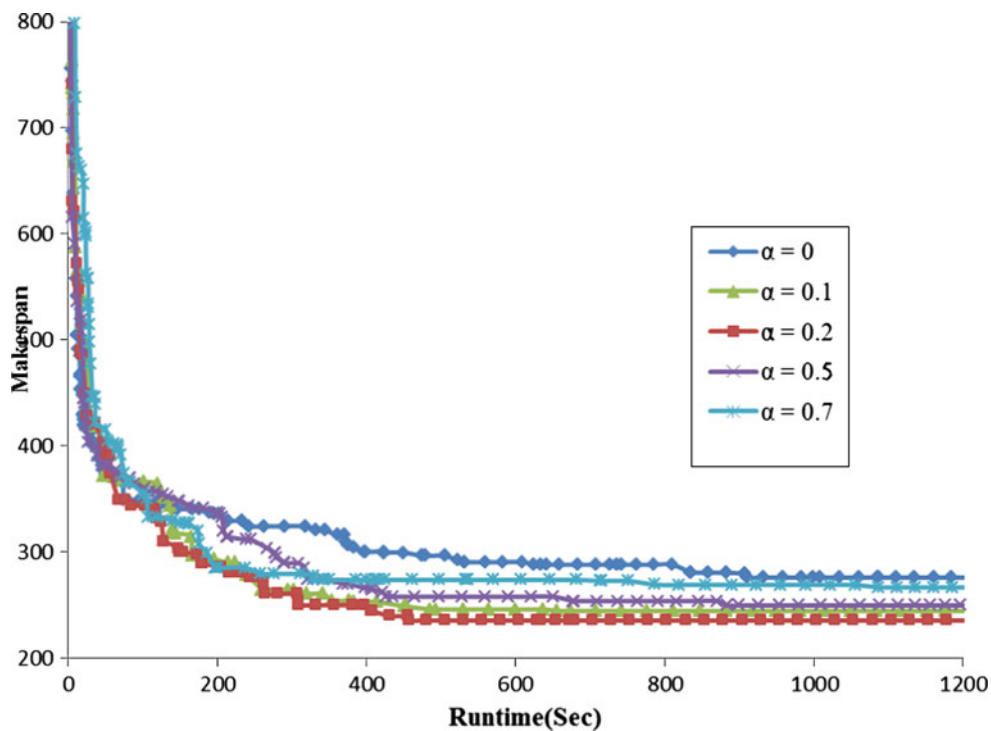


Fig. 8 Convergences conducted on different learning rates (WR5)

estimate the probability distribution. The conditional probability encoded in a network is used to generate the next set of sample solutions. The learning rate α affects the speed at

which the probability distribution is shifted to the promising data. It also affects the portions of the search space to be explored. For brevity, the discussion of the experiment result

analysis focuses only on the setting of the learning rate; the other parameters share the same configuration, as shown in Table 3.

The results in Fig. 7 show that α varies from 0.1 to 0.20, indicating that CEDA can achieve superior performance. If the learning rate value is too high, this may lead to premature convergence. Although the learning rate is too small, the algorithm may use more time to find the near optimal solution (shown in Fig. 8). CEDA degenerates into a co-evolutionary paradigm with a random search strategy when $\alpha = 0$, and it achieved superior results to traditional EAs.

Conclusion

This paper presents the proposed CEDA to overcome the challenges of modeling and evaluating complexity of S-MRS problems. First, CEDA applies a divide-and-conquer strategy to divide the overall problem into partitions, and each partition is mapped to a sub-population (species) following an evolutionary scheme. A positive result indicates that the CEDA approach is suitable for providing decision support in a dynamic operation environment, in which management preferences often change in response to the real-time conditions of the manufacturing system.

In our future work, further experiments will be conducted to determine the accuracy of the proposed CEDA in response to variations among the species of the ecosystem. The partnership, which includes the cooperators from each species as search points, is involved in CEDA. The construction and abandonment of the partnership are introduced briefly; however, the policy of abandonment and reconstruction must be further examined. Furthermore, we will extend CEDA to adapt to multiple-objective optimizations.

Acknowledgments This work was partly supported by the Grant-in-Aid for Scientific Research of Japan Society of Promotion of Science (C) (Grant No. 245102190001), the National Science Council, Taiwan (NSC101-2811-E-007-004; NSC100-2410-H-031-011-MY2; NSC100-2628-E-007-017-MY3), and the Advanced Manufacturing and Service Management Research Center of National Tsing Hua University (101N2073E1).

References

- Baluja, S. (1994). Population-based incremental learning: A method for integrating genetic search based function optimization and competitive, learning. CMU-CS-94-163.
- Baluja, S., & Caruana, R. (1995). Removing the genetics from the standard genetic algorithm. In *Proceedings of the twelfth international conference on machine learning* (pp. 38–46).
- Blazewicz, J., Lenstra, J. K., & Kan, A. H. G. R. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, 5(1), 11–24.
- Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1), 3–41.
- Chien, C.-F. (2007). Made in Taiwan: Shifting paradigms in high-tech industries. *Industrial Engineer*, 39(2), 47–49.
- Chien, C.-F., & Chen, C.-H. (2007a). Using genetic algorithms (GA) and a colored timed Petri net (CTPN) for modeling the optimization-based schedule generator of a generic production scheduling system. *International Journal of Production Research*, 45, 1763–1789.
- Chien, C.-F., & Chen, C.-H. (2007b). A novel timetabling algorithm for a furnace process for semiconductor fabrication with constrained waiting and frequency-based setups. *OR Spectrum*, 29(3), 391–419.
- Chien, C.-F. & Kuo, R.-T. (2012). Beyond make-or-buy: Cross-company short-term capacity backup in semiconductor industry ecosystem. *Flexible Services Manufacturing Journal*. doi:10.1007/s10696-011-9113-4. (in press).
- Chien, C.-F., & Zheng, J.-N. (2012). Mini-max regret strategy for robust capacity expansion decisions in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, 23(6), 2151–2159.
- Chien, C.-F., Chen, H.-K., Wu, J.-Z., & Hu, C.-H. (2007a). Constructing the OGE for promoting tool group productivity in semiconductor manufacturing. *International Journal of Production Research*, 45(3), 509–524.
- Chien, C.-F., Wang, H., & Wang, M. (2007b). A UNISON framework for analyzing alternative strategies of IC final testing for enhancing overall operational effectiveness. *International Journal of Production Economics*, 107(1), 20–30.
- Chien, C.-F., Chen, Y.-J., & Peng, J.-T. (2010a). Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle. *International Journal of Production Economics*, 128(2), 496–50.
- Chien, C.-F., Wu, J.-Z., & Weng, Y.-D. (2010b). Modeling order assignment for semiconductor assembly hierarchical outsourcing and its decision support system. *Flexible Services and Manufacturing Journal*, 22(1–2), 109–139.
- Chien, C.-F., Dauzère-Pérès, S., Ehm, H., Fowler, J. W., Jiang, Z., Krishnaswamy, Lee T.-E., et al. (2011a). Modeling and analysis of semiconductor manufacturing in a shrinking world: Challenges and successes. *European Journal of Industrial Engineering*, 5(3), 254–271.
- Chien, C.-F., Wu, J.-Z., & Wu, C.-C. (2011b). A two-stage stochastic programming approach for new tape-out allocation decisions for demand fulfillment planning in semiconductor manufacturing. *Flexible Services and Manufacturing Journal*. doi:10.1007/s10696-011-9109-0.
- Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. New York: Wiley-Interscience.
- Gen, M., Cheng, R., & Lin, L. (2008). *Network models and optimization: Multiobjective genetic algorithm approach*. Berlin: Springer.
- Gerwin, D. (1993). Manufacturing flexibility: A strategic perspective. *Management Science*, 39(4), 395–410.
- Guo, Y. W., Li, W. D., Mileham, A. R., & Owen, G. W. (2009). Optimisation of integrated process planning and scheduling using a particle swarm optimisation approach. *International Journal of Production Research*, 47(14), 3775–3796.
- Kim, Y. K., Kim, J. Y., & Shin, K. S. (2007). An asymmetric multileveled symbiotic evolutionary algorithm for integrated FMS scheduling. *Journal of Intelligent Manufacturing*, 18, 631–645.
- Li, W. D., & McMahon, C. A. (2007). A simulated annealing-based optimization approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 20(1), 80–95.
- Mönch, L., Fowler, J. W., Dauzère-Pérès, S., Mason, S. J., & Rose, O. (2011). A survey of problems, solution techniques, and future

- challenges in scheduling semiconductor manufacturing operations. *Journal of Scheduling*, 14(6), 583–599.
- Potter, M. A., & de Jong, K. A. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1), 1–29.
- Shih, W., Chien, C.-F., Shih, C., & Chang, J. (2009). The TSMC way: Meeting customer needs at Taiwan Semiconductor Manufacturing Co. *Harvard Business School Technology & Operations Management Unit Case* (610-003).
- Tan, K. C., Yang, Y. J., & Goh, C. K. (2006). A distributed cooperative coevolutionary algorithm for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 10(5), 527–549.
- Uzsoy, R., Lee, C. Y., & Martin-Vega, L. (1992). A review of production planning and scheduling models in the semiconductor industry, part I: System characteristics, performance evaluation, and production planning. *IIE Transactions*, 24, 47–60.
- Uzsoy, R., Martin-Vega, L., Lee, C., & Leonard, P. (1991). Production scheduling algorithms for a semiconductor test facility. *IEEE Transactions on Semiconductor Manufacturing*, 4(4), 270–280.
- Wu, J.-Z. (2011). Inventory write-down prediction for semiconductor manufacturing considering inventory age, accounting principle, and product structure with real settings. *Computers & Industrial Engineering*. doi:10.1016/j.cie.2011.11.020. (in press).
- Wu, J.-Z., & Chien, C.-F. (2008a). Modeling strategic semiconductor assembly outsourcing decisions based on empirical settings. *OR Spectrum*, 30(3), 401–430.
- Wu, J.-Z., & Chien, C.-F. (2008b). Modeling semiconductor testing job scheduling and dynamic modeling semiconductor testing job scheduling and dynamic testing machine configuration. *Expert Systems with Applications*, 35(1–2), 485–496.
- Wu, J.-Z., & Hsu, C.-Y. (2009). Critical success factors for improving decision quality on collaborative design in the IC supply chain. *Journal of Quality*, 16(2), 95–108.
- Wu, J.-Z., Chien, C.-F., & Gen, M. (2012a). Coordinating strategic outsourcing decisions for semiconductor assembly using a bi-objective genetic algorithm. *International Journal of Production Research*, 50(1), 235–260.
- Wu, J.-Z., Hao, X.-C., Chien, C.-F., & Gen, M. (2012b). A novel bi-vector encoding genetic algorithm for the simultaneous multiple resources scheduling problem. *Journal of Intelligent Manufacturing*, 23(6), 2255–2270.
- Zhang, W., & Gen, M. (2010). Process planning and scheduling in distributed manufacturing system using multiobjective genetic algorithm. *IEEE Transactions on Electrical and Electronic Engineering*, 5(1), 62–72.