OSA: Orthogonal Simulated Annealing Algorithm and Its Application to Designing Mixed H_2/H_∞ Optimal Controllers

Shinn-Jang Ho, *Member, IEEE*, Shinn-Ying Ho, *Member, IEEE*, and Li-Sun Shu, *Student Member, IEEE*

***Abstract*—This paper proposes a novel orthogonal simulated annealing (OSA) algorithm for solving intractable large-scale engineering problems and its application to designing mixed H_2/H_∞ optimal structure-specified controllers with robust stability and disturbance attenuation. High performance of OSA arises mainly from an intelligent generation mechanism (IGM), which applies orthogonal experimental design to speed up the search. IGM can efficiently generate a good candidate solution for next move of OSA by using a systematic reasoning method. It is difficult for existing H_∞- and genetic algorithm (GA)-based methods to economically obtain an accurate solution to the design problem of multiple-input, multiple-output (MIMO) optimal control systems. The high performance and validity of OSA are demonstrated by parametric optimization functions and a MIMO super maneuverable F18/HARV fighter aircraft system with a proportional-integral-derivative (PID)-type controller. It is shown empirically that OSA performs well for parametric optimization functions and the performance of the OSA-based method without prior domain knowledge is superior to those of existing H_∞- and GA-based methods for designing MIMO optimal controllers.**

***Index Terms*—Genetic algorithm (GA), H_2/H_∞ optimal control, orthogonal experimental design (OED), simulated annealing (SA).**

I. Introduction

A. Optimization Techniques

MANY intractable engineering problems, such as mixed H_2/H_∞ optimal control design [3], [4], [13], [15], are characterized by: 1) nonlinear multimodal search space; 2) large-scale search space; 3) tight constraints; and 4) expensive objective function evaluations. Therefore, it is desirable to develop an efficient optimization algorithm, such that accurate solutions can be economically obtained. The great success for evolutionary computation techniques, including evolutionary programming (EP), evolutionary strategies (ES), and genetic algorithm (GA), came in the 1980s when extremely complex optimization problems from various disciplines were solved, thus facilitating the undeniable breakthrough of evolutionary computation as a problem-solving methodology [1]. The evolutionary algorithm (EA) is a robust search and optimization methodology that is able to cope with ill-behaved problem domains, exhibiting attributes such as multimodality, discontinuity, time-variance, randomness, and noise [8].

The majority of control applications in the literature adopt the GA approach [8]. Recently, researchers have become increasingly interested in the use of GA as a means to design various classes of control systems [3], [4], [13], [15], [17]. GAs utilize a collective learning process of a population of individuals. Descendants are generated using randomized processes intended to model mutation and crossover. Mutation corresponds to an erroneous self-replication of individuals, while crossover exchanges information between two or more existing individuals. According to a fitness measure, the selection process favors better individuals to reproduce more often than those that are relatively worse [7]. The superiority of GA is achieved by using several search principles simultaneously such as population-based heuristics, and balance between global exploration and local exploitation.

To solve intractable engineering problems using GA, system parameters are encoded into individuals where each individual represents a search point in the space of potential solutions. A large number of system parameters would result in a very large search space. The performance of the conventional GA would be greatly degraded when applied to large parameter optimization problems that is shown by theoretical analysis in [14]. Furthermore, GAs have been shown to be efficient on global exploration by finding the most promising regions of the search space, but they suffer from excessively slow convergence to an accurate solution for tightly constrained problems with large-scale multimodal search spaces. This may prevent them from being really of practical interest for intractable large-scale engineering problems. Generally, GA with a local search heuristics is beneficial to improve the solution accuracy [22].

Simulated annealing (SA) is an efficient point-based optimization technique, which aims at escaping from local optima to find a globally optimal solution, and has been widely applied in various engineering problems [11], [19], [21], [29]. A standard SA algorithm consists of a sequence of iterations. Each iteration employs a randomized perturbation on a current solution, e.g., the mutation of GA, to generate a candidate solution in the neighborhood of the current solution. The neighborhood is defined by the choice of the generation mechanism. If the candidate solution is better than the current solution, it is accepted as the new current solution. Otherwise, it is accepted according to Metropolis's criterion

Manuscript received October 21, 2002; revised May 19, 2003, October 24, 2003, and March 4, 2004. This paper was recommended by Associate Editor Y. Pan.

S.-J. Ho is with the Department of Automation Engineering, National Huwei University of Science and Technology, Huwei, Yunlin, Taiwan 632, R.O.C. (e-mail: sjho@nhit.edu.tw).

S.-Y. Ho is with the Department of Biological Science and Technology and the Institute of Bioinformatics, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C. (e-mail: syho@mail.nctu.edu.tw).

L.-S. Shu is with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan 407, R.O.C. (e-mail: syho@fcu.edu.tw; p8600048@knight.fcu.edu.tw).

Digital Object Identifier 10.1109/TSMCA.2004.832834

1083-4427/04$20.00 © 2004 IEEE

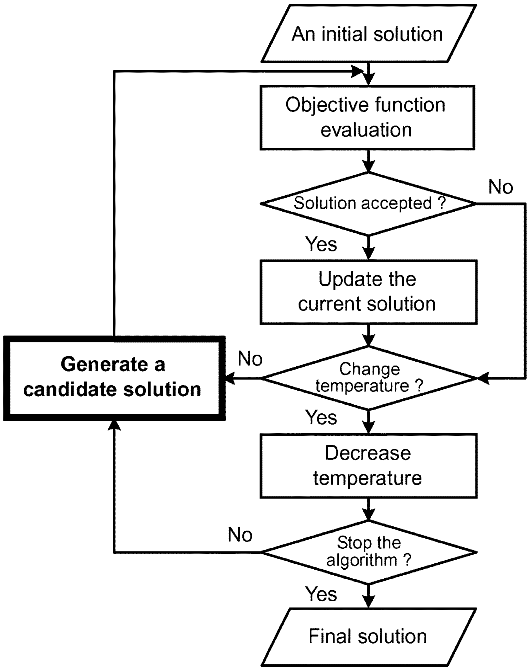

Fig. 1. Flowchart of a standard SA algorithm. The generation mechanism of OSA uses IGM, a systematic reasoning method based on orthogonal experimental design (OED), instead of the conventional generate-and-test method.

[18] based on Boltzman's probability. The flowchart of a standard SA algorithm is shown in Fig. 1. The generation mechanism of SA plays an important role in developing an efficient SA algorithm. The generation mechanism of the conventional SA using a generate-and-test method is difficult to explore an extremely large and nonlinear multimodal search space in a reasonable amount of computation time and is not acceptable for many intractable engineering applications [24], [26].

This paper proposes a novel orthogonal simulated annealing (OSA) algorithm for solving intractable large-scale engineering problems and its application to designing mixed $\mathrm{H}_2/\mathrm{H}_\infty$ optimal controllers with robust stability and disturbance attenuation without prior domain knowledge and differentiability assumption. OSA with an intelligent generation mechanism (IGM) can hybridize the advantages of global exploration and local exploitation by focusing on accuracy and computation time. IGM utilizes OED [2], [20], [28] to speed up the search and then can efficiently generate a good candidate solution for the next move by using a systematic reasoning method to efficiently exploit the neighborhood of a current solution instead of the generate-and-test method of the conventional SA, resulting in economically obtaining an accurate solution to the intractable engineering problem. It will be shown experimentally that OSA is superior to a number of efficient GAs [14] and a fast simulated annealing (FSA) algorithm [24] using parametric optimization functions.

OED with both orthogonal array (OA) and factor analysis is a representative method of quality control; it is also an efficient search mechanism. Tanaka proposed an orthogonal design algorithm ODA for a comparison with GA searching mechanisms [25]. ODA uses GA-encoding and OED, but uses no recombinations or mutations. OED can also be incorporated into the recombination operation of GA. Zhang and Leung proposed an orthogonal genetic algorithm OGA [30]. OGA divides each parent string into k parts, sample these parts from n parents based on the m combinations in an OA $L_m(n^k)$ to produce m binary strings, and then select j of them to be the offspring. Leung and Wang proposed an improved OGA with quantization (OGA/Q) using an OA-based initial population for global numerical optimization [16]. Both OGA and OGA/Q use OA, but use no factor analysis. Ho *et al.* proposed an EA with an OED-based recombination for efficiently solving large parameter optimization problems that the children are derived using both OA and factor analysis [10], [12]. The original contribution of this paper is to apply OED to SA rather than GAs and show superiority of the general-purpose OSA in solving large-scale parameter-optimization problems with real-world applications that is completely new with respect to published researches. The resultant OSA can be used to successfully design optimal mixed $\mathrm{H}_2/\mathrm{H}_\infty$ controllers.

B. Designing Optimal Mixed $\mathrm{H}_2/\mathrm{H}_\infty$ Controllers

Designing an optimal control system is equivalent to finding an optimal solution in a high-dimensional space, where each point represents a vector of design parameters. It is well recognized that the use of a large number of design parameters would result in a high-performance controller, provided that the used optimizer can obtain an accurate or even optimal solution to the optimal control problem. However, it is a usual way to alleviate the load of EAs by reducing the number of design parameters and consequently improve system performance by utilizing prior domain knowledge [3]. A survey of EAs in control system engineering can be found in [8].

In recent years, mixed $\mathrm{H}_2/\mathrm{H}_\infty$ optimal control problems have received a great deal of attention from the viewpoint of theoretical design [3], [4], [13], [15]. Chen *et al.* [4] used GA to design mixed $\mathrm{H}_2/\mathrm{H}_\infty$ optimal proportional-integral-derivative (PID) controllers, but the applied system is a single-input, single-output (SISO) system with few design parameters. Krohling and Rey [15] investigated the same problem using GA as [4] with a different performance index, time weighted square error for a short settling time. Chen and Cheng [3] used GA to design structure-specified multiple-input, multiple-output (MIMO) H_∞ optimal controllers for practical applications, but their procedure needs prior domain knowledge, i.e., the Routh–Hurwitz criterion for decreasing the domain size of each design parameter. Furthermore, to alleviate the load of GA, it is a usual way to reduce the number of design parameters by adopting the PI-type controller with 18 design parameters rather than the PID-type controller with 27 parameters [3], [13]. However, the performance of the GA-based method is worse than that of the conventional H_∞-based method [3]. Recently, Kitsios [13] used a GA-based method blended with multiobjective characteristics to improve the method of Chen and Cheng [3].

In this paper, a design problem of mixed $\mathrm{H}_2/\mathrm{H}_\infty$ optimal structure-specified MIMO controllers subject to two performance constraints: 1) robust stability and 2) disturbance attenuation is presented. Using OSA, lots of parameters can be efficiently tuned to obtain an accurate solution to the investigated problem. The high performance and validity of OSA are demonstrated by a MIMO super maneuverable F18/HARV

TABLE I
THE RESULTS OF A COMPLETE FACTORIAL EXPERIMENT. THE UNDERLINED NUMBERS OF H CORRESPOND TO A WELL-BALANCED SUBSET WHICH FORMS AN ORTHOGONAL ARRAY $L_9(3^4)$

h	***1***	2	3	4	***5***	6	7	8	***9***	10	***11***	12	13	14	***15***	***16***	17	18	19	20	***21***	***22***	23	24	25	***26***	27
x_1	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1
x_2	6	6	6	5	5	5	4	4	4	6	6	6	5	5	5	4	4	4	6	6	6	5	5	5	4	4	4
x_3	9	8	7	9	8	7	9	8	7	9	8	7	9	8	7	9	8	7	9	8	7	9	8	7	9	8	7
f_h	231	232	233	241	242	243	251	252	253	131	132	133	141	142	143	151	152	153	31	32	33	41	42	43	51	52	53
Rank of f_h	19	20	21	22	23	24	25	26	27	10	11	12	13	14	15	16	17	18	1	2	3	4	5	6	7	8	9

fighter aircraft system with a PID controller. Two cases of simulations are presented: one is a PI controller with 18 design parameters and the other is a PID controller with 27 design parameters. It will be shown empirically that the performance of the OSA-based PI controller is superior to those of existing H_∞- and GA-based controllers [3], [13]. Of course, the performance of the OSA-based PID controller is superior to the OSA-based PI controllers due to the increase in the number of design parameters and the superiority of OSA.

The remainder of this paper is organized as follows. Section II presents the OED used for IGM of OSA. Sections III and IV give the proposed IGM and OSA, respectively. The performance comparisons of OSA using parametric optimization functions are presented in Section V. Section VI describes the application of OSA to designing mixed H_2/H_∞ optimal controllers and gives performance comparisons with existing methods. Finally, Section VII concludes this paper.

II. OED

A. Concepts of OED

An efficient way to study the effect of several factors simultaneously is to use OED with both OA and factor analysis. The factors are the variables (parameters), which affect response variables, and a setting (or a discriminative value) of a factor is regarded as a level of the factor. A "complete factorial" experiment would make measurements at each of all possible level combinations. However, the number of level combinations is often so large that this is impracticable, and a subset of level combinations must be judiciously selected to be used, resulting in a "fractional factorial" experiment [5], [9]. OED utilizes properties of fractional factorial experiments to efficiently determine the best combination of factor levels to use in design problems.

An illustrative example of OED using an objective function is given as follows:

$$\min f(x_1, x_2, x_3) = 100x_1 - 10x_2 - x_3 \qquad (1)$$

where $x_1 \in \{1, 2, 3\}$, $x_2 \in \{4, 5, 6\}$, and $x_3 \in \{7, 8, 9\}$. This minimization problem can be regarded as an experimental design problem of three factors, with three levels each. Let factors 1–3 be parameters x_1, x_2, and x_3, respectively. Let the large, medial, and small values of each parameter be the levels 1–3 of each factor, respectively. The objective function f is the responsible variable. A complete factorial experiment would evaluate $3^3 = 27$ level combinations and then the best combination $(x_1, x_2, x_3) = (1, 6, 9)$ with $f = 31$ can be obtained. Let f_h denote an objective function value of the level combination h. The factorial array and results of the complete factorial experiment are shown in Table I. A fractional factorial experiment uses a well-balanced subset of level combinations, such as the 1st, 5th, 9th, 11th, 15th, 16th, 21st, 22nd, and 26th combinations. The best one of the nine combinations is the 21st combination $(x_1, x_2, x_3) = (1, 6, 7)$ with $f = 33$. Using OED, we can reason the best combination (1, 6, 9) from analyzing the results of the nine specific combinations, described in Section II-D.

OA is a fractional factorial array, which assures a balanced comparison of levels of any factor. OA is an array of numbers arranged in rows and columns, where each row represents the levels of factors in each combination, and each column represents a specific factor that can be changed from each combination. The term "main effect" designates the effect on response variables that one can trace to a design parameter [2]. The array is called orthogonal because all columns can be evaluated independently of one another, and the main effect of one factor does not bother the estimation of the main effect of another factor [5], [9].

Factor analysis using the OA's tabulation of experimental results can allow the main effects to be rapidly estimated, without the fear of distortion of results by the effects of other factors. Factor analysis can evaluate the effects of individual factors on the evaluation function, rank the most effective factors, and determine the best level for each factor such that the evaluation function is optimized.

OED can provide near-optimal quality characteristics for a specific objective. Furthermore, there is a large saving in the experimental effort. OED uses well-planned and controlled experiments in which certain factors are systematically set and modified, and the main effect of factors on the response can be observed. OED specifies the procedure of drawing a representative sample of experiments with the intention of reaching a sound decision [2]. Therefore, OED using OA and factor analysis is regarded as a systematic reasoning method.

B. OA

IGM uses one of two classes of OAs depending on applications. One is the class of two-level OAs used for optimization problems with a number of 0/1 decision variables [11].

TABLE II
ORTHOGONAL ARRAY $L_9(3^4)$

h	Factors 1	2	3	4	f_h
1	1	1	1	1	f_1
2	1	2	2	2	f_2
3	1	3	3	3	f_3
4	2	1	2	3	f_4
5	2	2	3	1	f_5
6	2	3	1	2	f_6
7	3	1	3	2	f_7
8	3	2	1	3	f_8
9	3	3	2	1	f_9

The other is the class of three-level OAs used for optimization problems with continuous/discrete parameters. All the optimization parameters are generally partitioned into N nonoverlapping groups. One group is regarded as a factor. In this study, the used three-level OA of IGM is described below.

Let there be N factors, with three levels each. The total number of level combinations is 3^N for a complete factorial experiment. To use an OA of N factors, we obtain an integer $M = 3^{\lceil \log_3(2N+1) \rceil}$ where the bracket represents an upper ceiling operation, build an OA $L_M(3^{(M-1)/2})$ with M rows and $(M-1)/2$ columns, use the first N columns, and ignore the other $(M-1)/2 - N$ columns. For example, if $N \in \{5, 6, \ldots, 13\}$, then $M = 27$ and $L_{27}(3^{13})$ is used. The numbers 1, 2, and 3 in each column indicate the levels of the factors. Each column has an equal number of 1s, 2s, and 3s. The array is orthogonal when the nine pairs, (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), and (3,3), appear the same number of times in any two columns. Table II illustrates an example of $L_9(3^4)$.

OA can reduce the number of level combinations for factor analysis. The number of OA combinations required to analyze all individual factors is only $M = O(N)$, where $2N+1 \leq M \leq 6N-3$. Algorithms of constructing OA's with various levels can be found in [16]. We give the algorithms for constructing the two- and three-level OAs used by OSA in the Appendix. After proper tabulation of experimental results, the summarized data are analyzed using factor analysis to determine the relative level effects of factors.

C. Factor Analysis

Define the main effect of factor i with level k as S_{ik} where $i = 1, \ldots, N$ and $k = 1, 2, 3$

$$S_{ik} = \sum_{h=1}^{M} f_h \cdot F_h \tag{2}$$

where $F_h = 1$ if the level of factor i of combination h is k; otherwise, $F_h = 0$. Consider the case that the objective function is to be maximized. The level k of factor i makes the best contribution to the objective function than the other two levels of factor i do when $S_{ik} = \max\{S_{i1}, S_{i2}, S_{i3}\}$. On the contrary, if the objective function is to be minimized, the level k is the best one when $S_{ik} = \min\{S_{i1}, S_{i2}, S_{i3}\}$. The main effect reveals the individual effect of a factor. The most effective factor has the largest one of main effect differences $\mathrm{MED}_i = \max\{S_{i1}, S_{i2}, S_{i3}\} - \min\{S_{i1}, S_{i2}, S_{i3}\}, i = 1, \ldots, N$.

After the best one of three levels of each factor is determined, an intelligent combination consisting of all factors with the best levels can be easily derived. OED is effective for development design of efficient search for the intelligent combination of factor levels, which can yield a high-quality objective function value compared with all values of generated combinations, and has a large probability that the reasoned value is superior to those of M representative combinations.

TABLE III
A CONCISE EXAMPLE OF IGM USING $L_9(3^4)$

h	Parameters x_1	x_2	x_3	f_h	Rank of f_h
1	3	6	9	231	19
2	3	5	8	242	23
3	3	4	7	253	27
4	2	6	8	132	11
5	2	5	7	143	15
6	2	4	9	151	16
7	1	6	7	33	3
8	1	5	9	41	4
9	1	4	8	52	8
S_{i1}	726	**396**	**423**		
S_{i2}	426	426	426		
S_{i3}	**126**	456	429		
Best level	3	1	1		
MED_i	600	60	6		
Solution x_i	1	6	9	31	1

Note that the main effect holds only when no or weak interaction exists, and that makes the experiment meaningful. An actual experiment result is estimated based only on the factors with the major effect. The difference between the estimated and experimental result is the degree of interactions among factors. In order to achieve an effective design, experiments should be prepared so as to avoid or reduce interactions.

D. Illustrative Example of OED

An illustrative example of OED for solving the optimization problem with (1) is described as follows (see Table III). First, use an $L_9(3^4)$, set levels for all factors as above mentioned, and evaluate the response variable f_h of the combination h, where $h = 1, \ldots, 9$. Second, compute the main effect S_{ik}, where $i = 1, 2, 3$ and $k = 1, 2, 3$. For example, $S_{21} = f_1 + f_4 + f_7 = 396$. Third, determine the best level of each factor based on the main effect. For example, the best level of factor 1 is level 3 since $S_{13} < S_{12} < S_{11}$. Therefore, select $x_1 = 1$. Finally, the best solution $(x_1, x_2, x_3) = (1, 6, 9)$ with $f = 31$ can be obtained. The most effective factor is x_1 with $MED_1 = 600$ which is the largest one. It can be verified from (1) that x_1 has the largest coefficient 100. Note that if only OA combinations without factor analysis are used, the obtained best solution is $(x_1, x_2, x_3) = (1, 6, 7)$ with $f = 33$, rather than the reasoned solution $(x_1, x_2, x_3) = (1, 6, 9)$.

III. IGM

Consider a parametric optimization function with p parameters and a current solution $X = [x_1, \ldots, x_p]^T$. IGM generates two temporary solutions $X_1 = \left[x_1^1, \ldots, x_p^1\right]^T$ and $X_2 = \left[x_1^2, \ldots, x_p^2\right]^T$ by perturbing X, where x_i^1 and x_i^2 are generated from x_i as follows:

$$x_i^1 = x_i + \overline{x}_i; \quad x_i^2 = x_i - \overline{x}_i, \qquad i = 1, \ldots, p. \tag{3}$$

The values of $\overline{x_i}$ are generated by the Cauchy–Lorentz probability distribution [24]. If x_i^1 $\left(x_i^2\right)$ is out of the domain range of x_i, randomly assign a feasible value to x_i^1 $\left(x_i^2\right)$. IGM aims at efficiently combining good values of parameters from solutions X, X_1, and X_2 to generate a good candidate solution Q for the next move.

Divide all the p parameters into N nonoverlapping groups with sizes l_i using the same division scheme for X, X_1, and X_2, $i = 1, \ldots, N$, such that

$$\sum_{i=1}^{N} l_i = p. \tag{4}$$

The proper value of N is problem-dependent. The larger the value of N, the more efficient the IGM is if the interactions among groups are weak. If the existing interactions among parameters are strong, the smaller the value of N, the more accurate the estimated main effect is. Considering the tradeoff, an efficient division criterion is to minimize the interactions among groups while maximizing the value of N. To efficiently use all columns of OA excluding the study of intractable interactions, the used OA is $L_{2N+1}(3^N)$ and the largest value of N is equal to $(3^{\lfloor \log_3(2p+1) \rfloor} - 1)/2$, where the bracket represents a lower ceiling operation. For example, if the interactions among $p = 41$ parameters are weak, the suggested value of $N = 40$ and the used OA is $L_{81}(3^{40})$. Note that the nest larger OA is $L_{243}(3^{121})$ with $N = 121$ which would waste at least $80 = 121 - 41$ columns of OA and results in a larger number $M = 243$ of function evaluations. The $N - 1$ cut points are randomly specified from the $p - 1$ candidate cut points which separate individual parameters. Note that the parameter N at each call of the following IGM operation can be a constant or variable value. For example, a coarse-to-fine strategy using a variable value of N is sometimes more efficient [12]. In the examples of this study, OSA uses a constant value of N.

How to perform an IGM operation with N groups using a current solution X with p parameters for an objective function f is described as follows:

- Step 1) Generate two temporary solutions X_1 and X_2 using X.
- Step 2) Divide each of X, X_1, and X_2 into N groups of parameters where each group is treated as a factor.
- Step 3) Use the first N columns of an OA $L_M(3^{(M-1)/2})$, where $M = 3^{\lceil \log_3(2N+1) \rceil}$.
- Step 4) Let levels 1–3 of factor i represent the ith groups of X, X_1, and X_2, respectively.
- Step 5) Compute f_h of the generated combination h, where $h = 2, \ldots, M$. Note that f_1 is the value of $f(X)$.
- Step 6) Compute the main effect S_{ik} where $i = 1, \ldots, N$ and $k = 1, 2, 3$.
- Step 7) Determine the best one of three levels of each factor based on the main effect.
- Step 8) The candidate solution Q is formed using the combination of the best groups.
- Step 9) Verify that Q is superior to the $M - 1$ sampling solutions derived from OA combinations and $Q \neq X$. If it is not true, select the best one of these $M - 1$ sampling solutions as the solution Q.

The number of objective function evaluations is M per IGM operation, which includes $M-1$ evaluations in Step 5 and one in Step 9. If interactions among groups are weak, Q is a potentially good approximation to the best one of all the 3^N combinations.

IV. OSA Algorithm

There are four choices must be made in implementing a SA algorithm for solving an optimization problem: 1) solution representation; 2) objective function definition; 3) design of the generation mechanism; and 4) design of a cooling schedule. The choices 1 and 2 are problem dependent. Designing an efficient generation mechanism plays an important role in developing SA algorithms. Generally, there are four parameters to be specified in designing the cooling schedule: 1) an initial temperature T_0; 2) a temperature update rule; 3) the number I of iterations to be performed at each temperature step; and 4) a stopping criterion of the SA algorithm.

The main power of OSA arises mainly from using IGM to efficiently search for a good candidate solution. OSA uses a simple geometric cooling rule by updating the temperature at the $(i+1)$th temperature step using the formula:, $T_{i+1} = \mathrm{CR} \cdot T_i$, $i = 0, 1, \ldots$ where CR is the cooling rate which is a constant smaller than 1 but close to 1. The higher the temperature, the larger it is the possibility of accepting the candidate solution worse than the current solution. OSA employs a variable value of I with an initial value I_0. The proper values of T_0, I_0, CR, and the stopping criterion are problem-dependent, generally specified by experienced engineers. Without loss of generality, consider the case that the objective function f is to be minimized. The proposed OSA is described as follows:

- Step 1) (Initialization) Initialize $T = T_0$, $I = I_0$, and CR. Randomly generate an initial solution X and compute $f(X)$. Let a counting variable $Count = 0$.
- Step 2) (Perturbation) Perform an IGM operation using X to generate a candidate solution Q.
- Step 3) (Acceptance criterion) Accept Q to be the new X with probability $P(Q)$

$$P(Q) = \begin{cases} 1, & \text{if } f(Q) \le f(X) \\ \exp\left(\frac{f(X)-f(Q)}{T}\right), & \text{if } f(Q) > f(X) \end{cases}. \tag{5}$$

- Step 4) (Iteration) Increase the value of $Count$ by one. If $Count < \lceil I \rceil$, go to Step 2.
- Step 5) (Decreasing temperature) Let the new values of T and I be $\mathrm{CR} \cdot T$ and $\mathrm{CR} \cdot I$, respectively. Reset $Count = 0$.
- Step 6) (Termination test) If a prespecified stopping criterion is met, stop the algorithm. Otherwise, go to Step 2.

OED has been proven optimal for additive and quadratic models, and the selected combinations are good representations for all of the possible combinations [28]. OA specifies a small number of representative combinations that are uniformly distributed over the neighborhood of the current solution. Furthermore, the factor analysis makes IGM more efficient in obtaining a good candidate solution which is a potentially good approximation to the best solution in the neighborhood.

TABLE IV
PERFORMANCE COMPARISONS OF VARIOUS ALGORITHMS FOR A PARAMETRIC FUNCTION WITH A GLOBAL OPTIMUM 121.598

	SGA	Stochastic GA	Sensitivity GA	GA-Local	SAGA	AGA	FSA	OSA
Chromosome length	1000 (Bit)	500 (Bit)	300 (Bit)	300 (Bit)	500 (Bit)	500 (Bit)	Real code	Real code
Population Size	51	31	51	51	51	51	1	1
Generation	750	200	NA	400	750	750	12,150	150
# of function evaluations	38,250	12,000	11,259	37,493	75,051	75,651	12,151	12,151
Function value	69	115	114	105	110.4	110.3	99.474	120.665

The overhead of IGM in preparing OA experiments and factor analysis is relatively small, compared with the cost of function evaluations. Note that the used OAs are generated in advance. Let G be the total number of iterations, which equals the number of IGM operations. The complexity of OSA is $GM+1$ function evaluations where M is the number of function evaluations per IGM operation. The complexity of the conventional SA, such as FSA [24], is $G+1$ function evaluations that an iteration takes one function evaluation. In other words, OSA and SA use $GM+1$ evaluations for G and GM iterations, respectively. When OSA is compared with SA using the same number of function evaluations, the actual computation time of OSA is generally much smaller than that of SA because OSA uses a smaller number of iterations.

V. PERFORMANCE COMPARISONS OF OSA

In this section, let the parameters of OSA for optimizing functions of p parameters be $T_0 = 50$, $I_0 = 5$, and $\mathrm{CR} = 0.95$. The used OA is $L_{2N+1}(3^N)$ where $N = (3^{\lfloor \log_3(2p+1) \rfloor} - 1)/2$. For comparisons, OSA and FSA [24] use the same prespecified number of function evaluations (N_{eval}) as the stopping criterion.

Since FSA takes more iterations than OSA using the same value of N_{eval}, FSA uses a greater value of CR than OSA. Therefore, FSA uses $T_0 = 50$, $\mathrm{CR} = 0.99$, and a constant value of $I = 5$.

A. Large Parameter Optimization Problem

KrishnaKumar *et al.* [14] proposed three approaches, stochastic GA, sensitivity GA, and GA-local search, and provided reasonable success on large parameter optimization problems using the test function $f(X)$ with $p = 100$ as follows:

$$\max f(X) = -\sum_{i=1}^{p} \left[\sin(x_i) + \sin\left(\frac{2x_i}{3}\right) \right] \tag{6}$$

where $X = [x_1, \ldots, x_p]^T$ and variable $x_i \in [3, 13]$. The performances of the three methods and a simple genetic algorithm (SGA) are cited from [14]. To demonstrate the high performance of OSA, OSA additionally compares with the following popular GAs: SAGA [6] and GA with adaptive probabilities of crossover and mutation (AGA) [23]. The average performances of all compared algorithms using 10 independent runs are shown in Table IV. This result reveals that OSA can efficiently obtain the best solution 120.665 using 0.3675 s. Note that FSA uses 0.7530 s which is much longer than that of OSA and the obtained solution is only 99.474. The second best solution is obtained by the stochastic GA.

TABLE V
BENCHMARK FUNCTIONS

Test functions	x_i domain	Optimum
$g_1 = -\sum_{i=1}^{p}\left[\sin(x_i) + \sin\left(\frac{2x_i}{3}\right)\right]$	[3, 13]	$1.21598p$ (max)
$g_2 = 6p + \sum_{i=1}^{p} \lfloor x_i \rfloor$	[-5.12, 5.12]	0 (min)
$g_3 = \sum_{i=1}^{p} \lfloor x_i + 0.5 \rfloor^2$	[-100, 100]	0 (min)
$g_4 = \sum_{i=1}^{p}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	[-5.12, 5.12]	0 (min)
$g_5 = -\sum_{i=1}^{p-1}\left[\sin(x_i + x_{i+1}) + \sin\left(\frac{2x_i x_{i+1}}{3}\right)\right]$	[3, 13]	$\approx 2p$ (max)
$g_6 = \frac{1}{4000}\sum_{i=1}^{p} x_i^2 - \prod_{i=1}^{p}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	0 (min)
$g_7 = \sum_{i=1}^{p-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	[-5.12, 5.12]	0 (min)
$g_8 = 20 + e - 20e^{-0.2\sqrt{\frac{\sum_{i=1}^{p} x_i^2}{p}}} - e^{\sum_{i=1}^{p}\frac{\cos(2\pi x_i)}{p}}$	[-30, 30]	0 (min)

B. Parametric Optimization Functions

To demonstrate the efficiency of OSA for solving optimization functions with various dimensions, eight benchmark functions gleaned from the literature, including unimodal and multimodal functions as well as continuous and discontinuous functions, are tested in the experimental study. The test function, variable domain, and global optimum for each function with p parameters are listed in Table V.

In order to show that the proposed IGM is effective, the simple OSA without heuristics is compared with FSA [24] and SGA with an elitist strategy using one-point crossover (ESGA) [7]. The parameters of ESGA are as follows: population size $= 50$, crossover rate $= 0.8$, mutation rate $= 0.005$. The simple ranking selection with the selection rate $=$ 0.2 is adopted, i.e., the worst two individuals are replaced with the best two individuals in a population. Each parameter is encoded using 10 bits for all test functions except that each parameter of g_6 uses 24 bits. The stopping criterion uses 10 000 function evaluations for all algorithms.

To illustrate the performance comparisons on various numbers of parameters, we use a distance function $\mathrm{Dist}(p)$ for describing the mean distance between the optimal solution $g_{\mathrm{opt}}(p)$ and the obtained best solution $g_{\mathrm{best}}(p)$ for one parameter as follows:

$$\mathrm{Dist}(p) = \frac{|g_{\mathrm{opt}}(p) - g_{\mathrm{best}}(p)|}{p}. \tag{7}$$

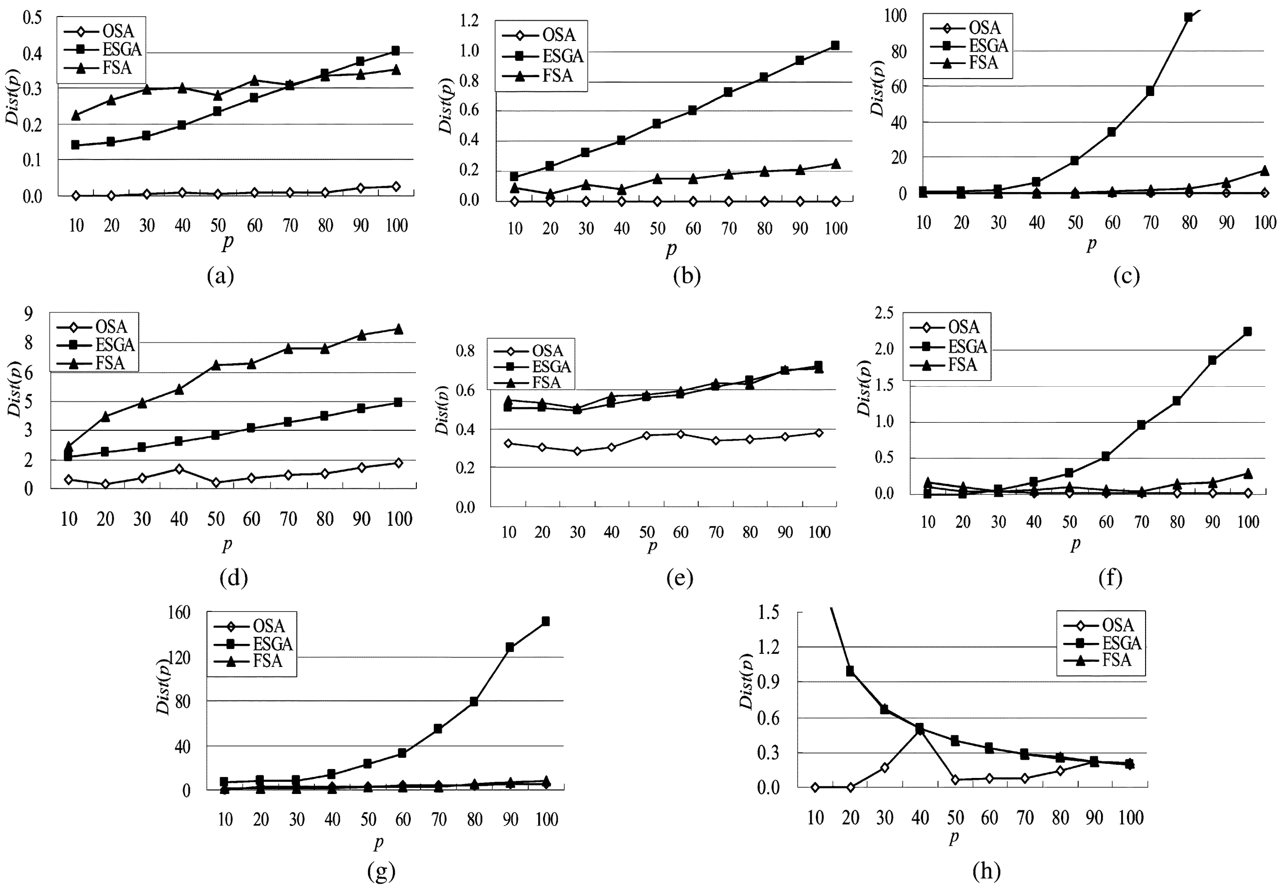

Fig. 2. Comparisons of various algorithms using $\mathrm{Dist(p)}$ curves for functions $g_1, g_2, \ldots,$ and g_8 in (a), (b),…, and (h), respectively.

TABLE VI
FITNESS VALUES AND RANKS FOR FUNCTIONS WITH $P = 10$ AND 100

Test functions	$p = 10$			$p = 100$		
	ESGA	FSA	OSA	ESGA	FSA	OSA
g_1	10.74546(2)	9.9078(3)	12.1598(1)	81.4246(3)	86.3480(2)	119.001(1)
g_2	1.5800(3)	0.8670(2)	0.0000(1)	103.4400(3)	24.9(2)	0.0000(1)
g_3	5.4333(3)	0.733(2)	0.0000(1)	14310.0(3)	1250.0(2)	17.36667(1)
g_4	16.5748(2)	21.443(3)	4.3778(1)	440.81(2)	823.91(3)	132.6545(1)
g_5	14.9670(2)	14.5390(3)	16.750(1)	128.0232(3)	129.33(2)	162.0802(1)
g_6	0.9996(2)	1.5970(3)	0.9995(1)	225.6208(3)	28.49(2)	1.1116(1)
g_7	72.81526(3)	9.03(2)	6.7751(1)	14989.0(3)	777.63(2)	495.6461(1)
g_8	19.9504(3)	19.615(2)	9.701×10^{-4}(1)	19.9504(2)	20.42(3)	19.7998(1)
Rank average	2.50	2.50	1.00	2.75	2.25	1.00
Final rank	2	2	1	3	2	1

The results of average $\mathrm{Dist}(p)$ for all test functions with $p = 10, 20, \ldots, 100$ using 30 independent runs are shown in Fig. 2 and Table VI. Fig. 2 reveals that the mean distance value $\mathrm{Dist}(p)$ of OSA slightly increases while p increases from 10 to 100, compared with other algorithms. From Fig. 2 and Table VI, it can be found that OSA outperforms ESGA and FSA for eight functions with various dimensions. This scenario reveals that OSA performs well in efficiently solving small and large parameter optimization problems in a limited amount of computation time.

VI. DESIGNING MIXED $\mathrm{H}_2/\mathrm{H}\infty$ OPTIMAL CONTROLLERS

A. Problem Description

The illustrative application of OSA is to effectively provide an accurate solution to the design problems of mixed $\mathrm{H}_2/\mathrm{H}_\infty$ optimal structure-specified controllers for systems with uncertainty and disturbance. The problem description is given as follows. Consider a MIMO control system with n_i inputs and n_o outputs as shown in Fig. 3, where $P(s)$ is the nominal plant, $\Delta P(s)$ is the plant perturbation, $C(s)$ is the controller, $r(t)$ is the reference input, $u(t)$ is the control input, $e(t)$ is the tracking

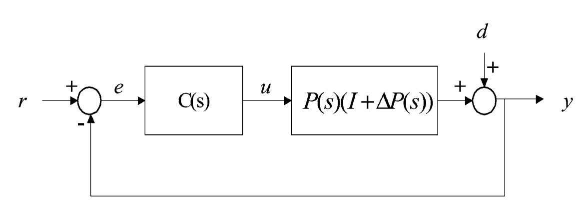

Fig. 3. Control system with plant perturbation and external disturbance.

error, $d(t)$ is the external disturbance, and $y(t)$ is the output of the system [3]. Without loss of generality, the plant perturbation $\Delta P(s)$ is assumed to be bounded by a known stable function matrix $W_1(s)$

$$\overline{\sigma}\big(\Delta P(jw)\big) \le \overline{\sigma}\big(W_1(jw)\big), \qquad \forall\, w \in [0, \infty) \tag{8}$$

where $\overline{\sigma}(\boldsymbol{A})$ denotes the maximum singular value of a matrix $\boldsymbol{A}$.

If a controller $C(s)$ is designed so that: 1) the nominal closed loop system ($\Delta P(s) = 0$ and $d(t) = 0$) is asymptotically stable and 2) the robust stability performance satisfies the following inequality:

$$J_a = \|W_1(s)T(s)\|_\infty < 1 \tag{9}$$

and 3) the disturbance attenuation performance satisfies the following inequality:

$$J_b = \|W_2(s)S(s)\|_\infty < 1 \tag{10}$$

then the closed loop system is also asymptotically stable with $\Delta P(s)$ and $d(t)$. Where $W_2(s)$ is a stable weighting function matrix specified by designers. $S(s)$ and $T(s) = I - S(s)$ are the sensitivity and complementary sensitivity functions of the system, respectively

$$S(s) = \big(I + P(s)C(s)\big)^{-1} \tag{11}$$

$$T(s) = P(s)C(s)\big(I + P(s)C(s)\big)^{-1} \tag{12}$$

and the H_∞-norm in (9) and (10) is defined as

$$\|A(s)\|_\infty \equiv \max_{\omega} \overline{\sigma}\big(A(j\omega)\big). \tag{13}$$

A balanced performance criterion to minimize both J_a and J_b simultaneously is to minimize J_∞ [3], [13]

$$J_\infty = \big(J_a^2 + J_b^2\big)^{1/2}. \tag{14}$$

For advancing the system performance, robust stability and disturbance attenuation are often not enough in the control system design. The minimization of tracking error J_2 (i.e., H_2 norm) should be taken into account

$$J_2 = \int_0^\infty e^T(t)e(t)dt \tag{15}$$

where $e(t) = r(t) - y(t)$ is the error which can be obtained from the inverse Laplace transformation of $E(s)$ with $\Delta P(s) = 0$ and $d(t) = 0$

$$E(s) = \big(I + P(s)C(s)\big)^{-1} R(s). \tag{16}$$

In the proposed method, the handling of constraints (9) and (10) is to recast the constraints as objectives to be minimized and, consequently, a weighted-sum approach is conveniently used. Therefore, the objective function of the investigated problem of designing mixed H_2/H_∞ optimal controllers is as follows:

$$\min_C J = J_2 + J_\infty. \tag{17}$$

The order of the derived optimal controller is very high by using conventional methods, so that it is not easy to be implemented. To cope with this difficulty, we investigate the mixed H_2/H_∞ optimal control problem from suboptimal perspective. A structure-specified controller of the following form [3]:

$$C(s) = \frac{N_c(s)}{D_c(s)} = \frac{\boldsymbol{B}_m s^m + \boldsymbol{B}_{m-1}s^{m-1} + \cdots + \boldsymbol{B}_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0} \tag{18}$$

is assigned with some desired orders m and n to minimize J, where

$$\boldsymbol{B}_k = \begin{bmatrix} b_{k11} & \cdots & b_{k1n_i} \\ \vdots & \ddots & \vdots \\ b_{kn_o1} & \cdots & b_{kn_on_i} \end{bmatrix} \tag{19}$$

for $k = 0, 1, \ldots, m$. Most of the conventional controllers used in industrial control systems have fundamental structures such as PID and lead/lag configurations. Such controllers are special cases of the structure-specified controllers. For the PID controller, we have $n = 1$, $m = 2$ and $a_0 = 0$, i.e.,

$$C(s) = \frac{\boldsymbol{B}_2 s^2 + \boldsymbol{B}_1 s + \boldsymbol{B}_0}{s}$$

$$\boldsymbol{B}_k = \begin{bmatrix} b_{k11} & b_{k12} & b_{k13} \\ b_{k21} & b_{k22} & b_{k23} \\ b_{k31} & b_{k32} & b_{k33} \end{bmatrix}, \qquad k = 0, 1, 2. \tag{20}$$

The PID controller has 27 design parameters. A PI controller with 18 design parameters is a special case of the PID controller, where $\boldsymbol{B}_2 = 0$. Similarly, for the lead/lag controller, we have $n = m = 1$, i.e.,

$$C(s) = \frac{\boldsymbol{B}_1 s + \boldsymbol{B}_0}{s + a_0}. \tag{21}$$

B. Design of Controllers Using OSA

For convenience and simplicity, from the structure-specified controller (18), we denote

$$\begin{aligned}\theta &= [a_0 \cdots a_{n-1} b_{011} \cdots b_{01n_i} b_{021} \cdots b_{02n_i} \cdots b_{mn_on_i}]^T \\ &= [\theta_1, \ldots, \theta_p]^T\end{aligned} \tag{22}$$

as the solution representation, where $p = n + (m+1) \times n_i \times n_o$ is the number of design parameters. Denote Θ as the search space consisting of all admissible θ_i, $i = 1, \ldots, p$. The structure-specified mixed H_2/H_∞ optimal control design problem is equivalent to finding an optimal θ from Θ to minimize the objective function J in (17) subject to the inequality constraints (9) and (10). Chen and Cheng [3] used prior domain knowledge, i.e., the Routh–Hurwitz criterion, for decreasing the domain size of each design parameter θ_i. In this study, we do not use any domain knowledge to confine the search space Θ in order to demonstrate the strong search ability of OSA in efficiently obtaining a near-optimal solution to the investigated problem.

The parameters of OSA, using a constant value of $I = 1$ for the investigated problem, are: N, CR, T_0, and threshold values δ and N_{stop} for specifying a stopping condition. The proposed OSA-based method for finding a near-optimal solution θ_{opt} to the mixed H_2/H_∞ optimal control design problem is described as follows.

- Step 1) Randomly generate a solution θ as a current solution and let $\theta_{\text{opt}} = \theta$. Let J^i be the value of J at the ith iteration and $i = 0$. Compute the value J^0.
- Step 2) Perform an IGM operation using θ to generate a candidate solution Q.
- Step 3) Accept Q to be the new θ with probability $P(Q)$ in (5).
- Step 4) Increase i by one. Compute J^i using the current θ. If $J^i < J^{i-1}$, let the new θ_{opt} be θ.

Step 5) Decrease temperature using. $T_{i+1} = CR \cdot T_i$.
Step 6) Let $\Delta J^i = |J^{i-1} - J^i|/J^{i-1}$. If $\Delta J^k \leq \delta$ for $k = i, i-1, \ldots, i-(N_{\text{stop}}+1)$, stop the algorithm. Otherwise, go to Step 2.

C. Test Problem

For comparison with the GA-based methods proposed in [3] and [13], the same MIMO optimal control design problem is tested. Consider the design problem of a longitudinal control system of the super maneuverable F18/HARV fighter aircraft in horizontal flight at an altitude of 15 000 (ft) with Mach number 0.24, airspeed $V_T = 238.7$ (ft/s), attack angle $\alpha = 25$ (°), and pitch angle $\beta = 25$ (°). The trim value of the path angle is $\beta - \alpha = 0$ (°) and the trim pitch rate is $\gamma = 0$ (°/s). The longitudinal dynamics of the system can be described as

$$\left.\begin{aligned} \dot{x} &= \boldsymbol{A}x + \boldsymbol{B}u \\ y &= \boldsymbol{C}x \end{aligned}\right\} \tag{23}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are given as shown in (24) at the bottom of the page, $x = [V_T, \alpha, \gamma, \beta]^T$ and $u = [u_{TV}, u_{AS}, u_{SS}, u_{LE}, u_{TE}, u_T]^T$. Where u_{TV}, u_{AS}, u_{SS}, u_{LE}, u_{TE}, and u_T are the perturbations in symmetric thrust vectoring vane deflection, symmetric aileron deflection, symmetric stabilator deflection, symmetric leading edge flap deflection, symmetric trailing edge flap deflection, and throttle position, respectively. Note that the rank of the matrix $\boldsymbol{B}$ is only three. It is important to remove the redundancy in the control inputs. By employing the pseudocontrol technique [27], we can transform the six control inputs (u_{TV}, u_{AS}, u_{SS}, u_{LE}, u_{TE}, and u_T) to three linearly independent variables. Therefore, the system can be rewritten as

$$\dot{x} = \boldsymbol{A}x + \boldsymbol{B}_v v \tag{25}$$

where $\boldsymbol{B}_v$ and v are given as shown in (26) at the bottom of the next page.

Suppose the reference input is $r(t) = [0, 1-e^{-3t}, 1-e^{-6t}]^T$ and the system is encountering with the external disturbance $d(t) = 0.01e^{-0.2t}\cos(3162.3t)\,[1,1,1]^T$. The bound $W_1(s)$ of the plant perturbation $\Delta P(s)$ is

$$W_1(s) = \frac{0.0125s^2 + 1.2025s + 1.25}{s^2 + 20s + 100} I_{3\times 3}. \tag{27}$$

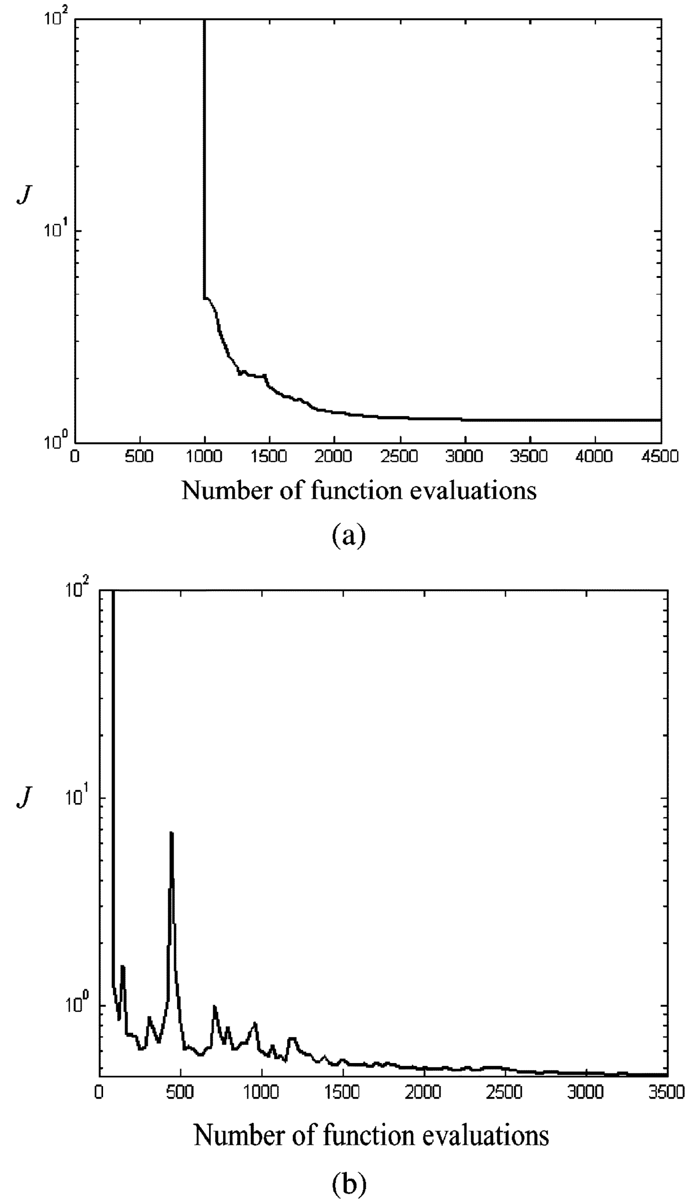

Fig. 4. Average convergence of OSA from tens runs. (a) PI controller. (b) PID controller.

$$\boldsymbol{A} = \begin{bmatrix} -0.0750 & -24.0500 & 0 & -36.1600 \\ -0.0009 & -0.1959 & 0.9896 & 0 \\ -0.0002 & -0.1454 & -0.1677 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$
$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\boldsymbol{B} = \begin{bmatrix} -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \\ -0.0002 & -0.0001 & -0.0004 & 0 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & 0.0007 & 0.0005 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{24}$$

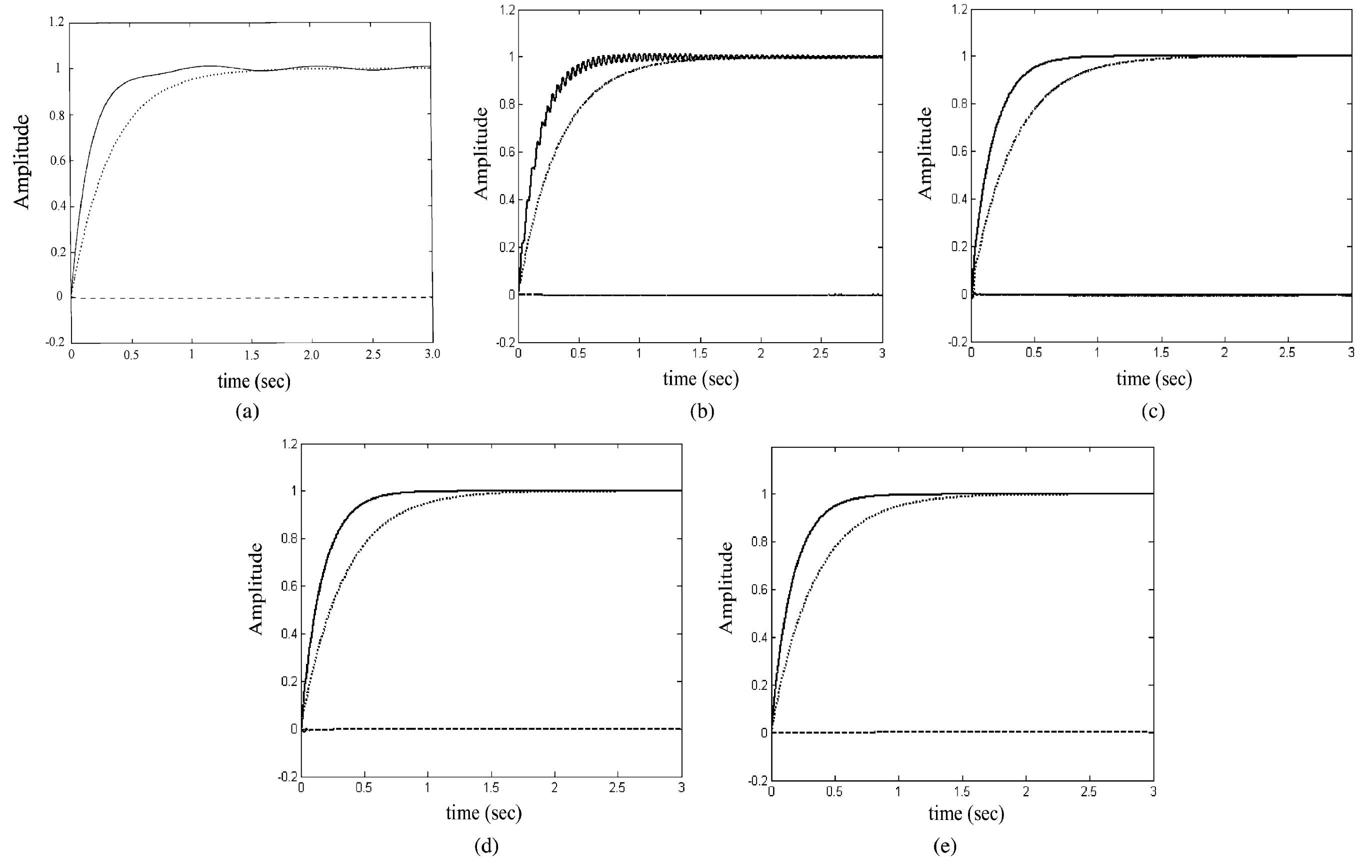

Fig. 5. Outputs of systems using various controllers. (a) H_∞-based controller [3]. (b) GA-based PI controller [3]. (c) Improved GA-based PI controller [13]. (d) OSA-based PI controller. (e) OSA-based PID controller.

To attenuate disturbance, the stable weighting function $W_2(s)$ consisting of a low-pass filter is chosen as

$$W_2(s) = \frac{0.25s + 0.025}{s^2 + 0.4s + 10\,000\,000} I_{3\times 3}. \tag{28}$$

The search space Θ consists of all admissible $\theta_i \in [-20000, 20\,000]$, $i = 1, \ldots, p$ [3], [13].

D. Experimental Results

Let the parameters of OSA be $N = 13, CR = 0.95, T_0 = 80$, and $N_{\text{stop}} = 3$. The stopping criteria of OSA for PI and PID controllers are $\delta = 9 \times 10^{-3}$ and $\delta = 10^{-4}$, respectively. Ten

TABLE VII
PERFORMANCE COMPARISONS IN TERMS OF J_2 AND J_∞ FOR VARIOUS CONTROLLERS

Controller	J_2	J_∞	$J=J_2+J_\infty$	N_{eval}
GA-based PI controller [3]	NA	0.8194	NA	18,000
GA-based PI controller [13]	0.1114	0.7682	0.8796	4,500
OSA-based PI controller	0.0374	0.6299	0.6673	3,781
OSA-based PID controller	0.0019	0.4374	0.4393	2,971

$$\boldsymbol{B}_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$v = \begin{bmatrix} -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \\ -0.0002 & -0.0001 & -0.0004 & 0 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & 0.0007 & 0.0005 \end{bmatrix} u \tag{26}$$

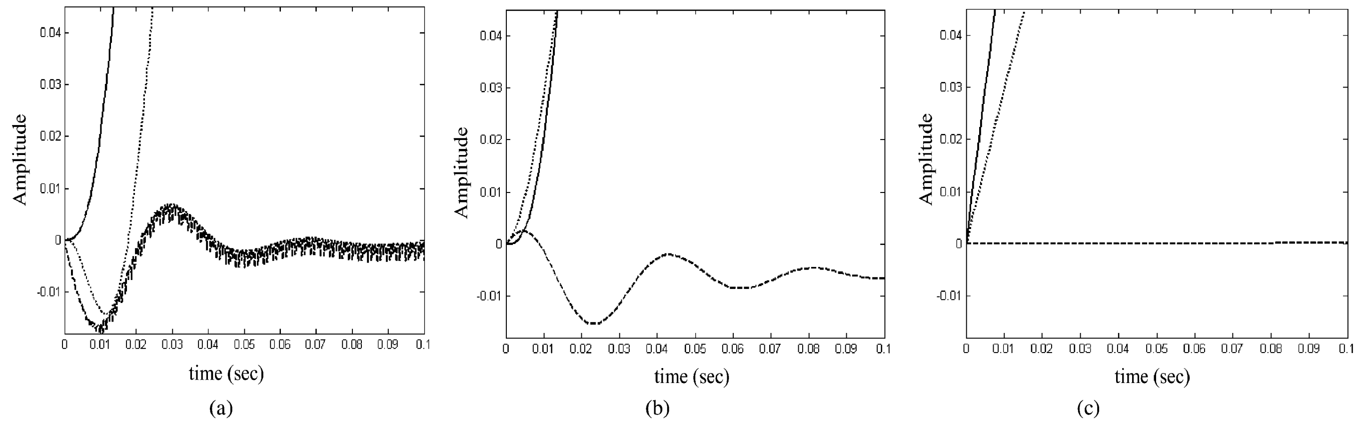

Fig. 6. Detailed tracking response of the first 0.1-s period. (a) The improved GA-based PI controller [13]. (b) OSA-based PI controller. (c) OSA-based PID controller.

independent runs are conducted for each of PI and PID controllers. The best PI controller with $J = 0.6673$ is obtained using 140 iterations, 3781 function evaluations as follows:

$$C(s) = \frac{\begin{bmatrix} 2055.94 & 1373.15 & -103.66 \\ -1184.03 & -1038.37 & 65.41 \\ -15\,397.39 & 8715.97 & 19\,989.70 \end{bmatrix} s}{s} + \frac{\begin{bmatrix} 19\,620.59 & 2995.98 & -3744.34 \\ 14\,513.74 & -19\,999.97 & -14\,359.620 \\ 3102.22 & -19\,810.43 & 13\,970.38 \end{bmatrix}}{s}. \quad (29)$$

The average convergence of OSA for PI controllers is shown in Fig. 4(a). The best PID controller with $J = 0.4393$ is obtained using 110 iterations, 2971 function evaluations as follows:

$$C(s) = \frac{\begin{bmatrix} -19\,788.38 & -18\,237.71 & -1347.88 \\ -12\,517.98 & 8483.12 & 622.55 \\ 11\,498.13 & 917.12 & 3375.39 \end{bmatrix} s^2}{s} + \frac{\begin{bmatrix} 19\,185.69 & -12\,519.68 & 19\,960.12 \\ -20\,000.00 & 7084.61 & -18\,786.01 \\ -10\,107.62 & 19\,360.44 & 5944.56 \end{bmatrix} s}{s} + \frac{\begin{bmatrix} -19\,060.26 & 18\,026.98 & 20\,000.00 \\ 4194.82 & -2302.57 & 6856.19 \\ -5180.15 & 11\,857.10 & -2348.46 \end{bmatrix}}{s}. \quad (30)$$

The average convergence of OSA for PID controllers is shown in Fig. 4(b). This OSA-based PI and PID controllers are applied to the control system to illustrate the high performance of the proposed method. The outputs of the system of the derived controllers with robust stability and disturbance attenuation are shown in Fig. 5.

The outputs of the systems using the GA-based PI controllers and conventional H_∞-based controller for the same test problem are borrowed from [3] and [13], as shown in Fig. 5. The GA-based method [3] used the population size $N_{\text{pop}} = 100$, crossover rate $p_c = 0.9$, mutation rate $p_m = 0.2$, and 200 generations. The improved GA-based method [13] used $N_{\text{pop}} = 100$, $p_c = 0.9$, $p_m = 0.02$, and 50 generations. The performance comparisons in terms of J_2 and J_∞ for various controllers are given in Table VII. Note that only the H_∞ norm value J_∞ without tracking error J_2 is considered in the PI controllers of the GA-based methods [3] and [13]. The value of J_2 of the method [13] is derived from the reported controller and that of the method [3] is not available. Fig. 5(a) and (b) reveal that the H_∞-based controller is superior to the GA-based controller [3] because the simple GA is hard to cope with large parameter optimization problems for obtaining an accurate solution in a limited amount of computation time. Fig. 5(c) shows that the improved GA-based method [13] performs better than the H_∞-based method. However, the performance of the proposed OSA-based PI controller is superior to those of the GA- and H_∞-based controllers from the comparisons of Fig. 5 and Table VII. Of course, the performance of the OSA-based PID controller is superior to the OSA-based PI controllers due to the increase in the number of design parameters and the superiority of OSA.

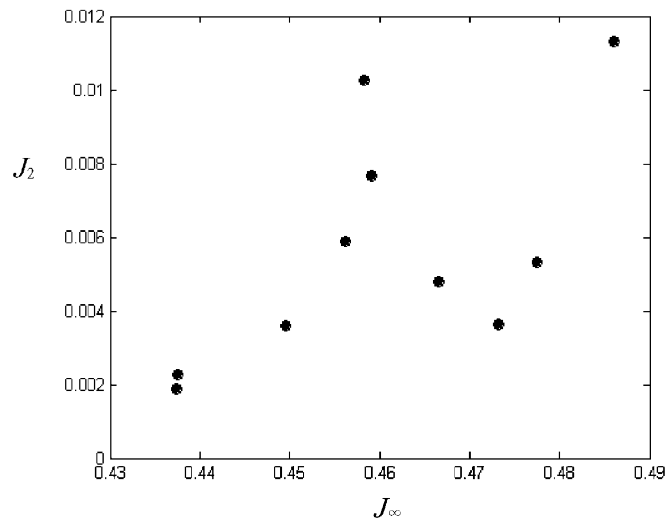

Fig. 7. Distribution of the ten PID controllers.

The encouraging performance, $J_2 = 0.0019$ and $J = 0.4374$, of the PID controller using 2971 function evaluations demonstrate the efficiency of the OSA-based method. To further examine the performance of the system outputs in Fig. 5(c)–(e), we show the detailed tracking response of the first 0.1-s period in Fig. 6. It is obvious that the OSA-based PI controller has smaller oscillations than the improved GA-based one [13] and

TABLE VIII
PERFORMANCE OF THE OSA-BASED PID CONTROLLER FROM TEN INDEPENDENT RUNS

Controller	J_2			J_∞			J		
	best	*avg.*	*std.*	*best*	*avg.*	*std.*	*best*	*avg.*	*std.*
OSA-based PID controller	0.0019	0.0057	0.0032	0.4374	0.4601	0.0161	0.4393	0.4658	0.0183

the PID controller has almost no oscillation. The 10 OSA-based PID controllers are reported in Fig. 7 and Table VIII. The standard deviations of J_2 and J_∞ are very small. The simulation results illustrate that the OSA-based method can economically and robustly provide a near-optimal solution to the problem of designing mixed H_2/H_∞ optimal structure-specified MIMO controllers with robust stability and disturbance attenuation without using prior domain knowledge.

VII. CONCLUSION

This paper proposes an OSA algorithm and its application to providing a near-optimal solution to the problem of designing mixed H_2/H_∞ optimal structure-specified MIMO controllers with robust stability and disturbance attenuation. OSA performs well in solving intractable engineering problems comprising lots of system parameters. The IGM of OSA can adaptively adopt the two- and three-level OAs for handling various optimization problems. It is shown that OSA outperforms conventional GAs and SA for high-dimensional parametric optimization functions in a limited amount of computation time. The optimal control design problem is to minimize the tracking error (H_2-norm) with robust constraints of the type H_∞-norm. The OSA-based method without prior domain knowledge can efficiently solve the design problems of MIMO optimal control systems. The high performance and validity of the proposed method are demonstrated by a MIMO super maneuverable F18/HARV fighter aircraft system with PI and PID controllers. It is shown empirically that the performance of the proposed method is much superior to those of existing GA- and H_∞-based methods. The OSA-based method can be widely used for designing high-performance optimal controllers. We believe that domain knowledge and auxiliary techniques can further advance the performance of the OSA-based method in solving various engineering problems.

APPENDIX

The following algorithm generates the Q-level OA $L_M(Q^K)$ used by OSA with N factors where $M = Q^J$ and $K = (M-1)/(Q-1)$, $Q = 2, 3$. If $Q = 2$, $J = \lceil \log_2(N+1) \rceil$. If $Q = 3$, $J = \lceil \log_3(2N+1) \rceil$. Let $\boldsymbol{a}_j$ be the jth column of the OA $[a_{i,j}]_{M\times K}$. The columns $\boldsymbol{a}_j$ where $j = 1, 2, (Q^{2-1})/(Q-1)+1, (Q^3-1)/(Q-1)+1, \ldots, (Q^{J-1}-1)/(Q-1)+1$ are called basic columns, and the others are called nonbasic columns.

```
Step 1: Construct the basic columns.
 for k = 1 to J do
  {
  j = (Q^(k-1) - 1)/(Q - 1) + 1;
  for i = 1 to M do
   a_{i,j} = ⌊(i - 1)/Q^(J-k)⌋ mod Q
 }
Step 2: Construct the nonbasic columns.
 for k = 2 to J do
 {
  j = (Q^(k-1) - 1)/(Q - 1) + 1;
  for s = 1 to j - 1 do
   for t = 1 to Q - 1 do
    a_{j+(s-1)(Q-1)+t} = (a_s × t + a_j) mod Q;
 }
Step 3: Increase a_{i,j} by one where i =
1,...,M and j = 1,...,K.
```

REFERENCES

[1] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. New York: Oxford Univ. Press, 1997.

[2] T.-P. Bagchi, *Taguchi Methods Explained: Practical Steps to Robust Design*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[3] B.-S. Chen and Y.-M. Cheng, "A structure-specified H_∞(optimal control design for practical application s: A genetic approach," *IEEE Trans. Contr. Syst. Technol.*, vol. 6, pp. 707–718, Nov. 1998.

[4] B.-S. Chen, Y.-M. Cheng, and C.-H. Lee, "A genetic approach to mixed H_2/H_∞ optimal PID control," *IEEE Contr. Syst. Mag.*, vol. 15, pp. 51–60, Oct. 1995.

[5] A. Dey, *Orthogonal Fractional Factorial Designs*. New York: Wiley, 1985.

[6] H. Esbensen and P. Mazumder, "SAGA: A unification of the genetic algorithm with simulated annealing and its application to macro-cell placement," in *Proc. IEEE Int. Conf. VLSI Design*, 1994, pp. 211–214.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[8] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: A survey," *Contr. Eng. Practice*, vol. 10, pp. 1223–1241, 2002.

[9] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal Arrays: Theory and Applications*. New York: Springer-Verlag, 1999.

[10] S.-Y. Ho, C.-C. Liu, and S. Liu, "Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm," *Pattern Recognition Lett.*, vol. 23, pp. 1495–1503, 2002.

[11] S.-Y. Ho, S.-J. Ho, and Y.-K. Lin, "An orthogonal simulated annealing for floorplanning problems," *IEEE Trans. VLSI Syst.*, vol. 12, Aug. 2004.

[12] H.-L. Huang and S.-Y. Ho, "Mesh optimization for surface approximation using an efficient coarse-to-fine evolutionary algorithm," *Pattern Recognit.*, vol. 36, no. 5, pp. 1065–1081, 2003.

[13] I. Kitsios, T. Pimenides, and P. Groumpos, "A genetic algorithm for designing H_∞ structured specified controllers," in *Proc. IEEE Int. Conf. Contr. Applicat.*, Mexico, 2001, pp. 1196–1201.

[14] K. KrishnaKumar, S. Narayanaswamy, and S. Garg, *Solving Large Parameter Optimization Problems Using a Genetic Algorithm With Stochastic Coding*, G. Winter, J. Périaux, M. Galaán, and P. Cuesta, Eds. New York: Wiley, 1995, pp. 287–303.

[15] R.-A. Krohling and J.-P. Rey, "Design of optimal disturbance rejection PID controllers using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 5, pp. 78–82, Feb. 2001.

[16] Y.-W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 5, pp. 41–53, Feb. 2001.

[17] M.-A. Marra, B.-E. Boling, and B.-L. Walcott, "Genetic control of a ball-beam system," in *Proc. IEEE Int. Conf. Control Applicat.*, 1996, pp. 608–613.

[18] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[19] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence pair," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 1518–1518, Dec. 1996.

[20] M.-S. Phadke, *Quality Engineering Using Robust Design*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[21] T. Renyuan, S. Jianzhong, and L. Yan, "Optimization of electromagnetic devices by using intelligent simulated annealing algorithm," *IEEE Trans. Magn.*, vol. 34, pp. 2992–2995, Sept. 1998.
[22] H. Sayoud, K. Takahashi, and B. Vaillant, "A genetic local tuning algorithm for a class of combinational networks design problems," *IEEE Commun. Lett.*, vol. 5, pp. 322–324, July 2001.
[23] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 656–667, Apr. 1994.
[24] H. Szu and R. Hartley, "Fast simulated annealing," *Phys. Lett.*, vol. 122, pp. 157–162, 1987.
[25] H. Tanaka, "A comparative study of GA and orthogonal experimental design," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1997, pp. 143–146.
[26] X. Tang, R. Tian, and D. F. Wong, "Fast evaluation of sequence pair in block placement by longest common subsequence computation," *IEEE Trans. Computer-Aided Design*, vol. 20, pp. 1406–1406, Dec. 2001.
[27] P. Voulgaris and L. Valavani, "High performance H_2 and H_∞ designs for supermaneuverable F18/harv fighter aircraft," *AIAA J. Guid. Cont. Dyn.*, vol. 14, no. 1, pp. 157–165, 1991.
[28] Q. Wu, "On the optimality of orthogonal experimental design," *Acta Math. Appl. Sinica*, vol. 1, no. 4, pp. 283–299, 1978.
[29] S. Yang, J.-M. Machado, G. Ni, S.-L. Ho, and P. Zhou, "A self-learning simulated annealing algorithm for global optimizations of electromagnetic devices," *IEEE Trans. Magn.*, vol. 36, pp. 1004–1008, July 2000.
[30] Q. Zhang and Y.-W. Leung, "An orthogonal genetic algorithm for multimedia multicast routing," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 53–62, Apr. 1999.

Shinn-Jang Ho (M'03) was born in Taiwan, R.O.C., in 1962. He received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1984, 1986, and 1992, respectively.

From 1992 to 2004, he was with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan, R.O.C. He is currently a Professor in the Department of Biological Science and Technology and the Institute of Bioinformatics, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. His research interests include evolutionary algorithms, image processing, pattern recognition, bioinformatics, data mining, virtual reality applications of computer vision, fuzzy classifier, large parameter optimization problems, and system optimization.

Shinn-Ying Ho (M'00) was born in Taiwan, R.O.C., in 1962. He received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1984, 1986, and 1992, respectively.

He is currently a Professor in the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His research interests include image processing, pattern recognition, virtual-reality applications of computer vision, fuzzy classifier, evolutionary algorithms, large parameter optimization problems, and system optimization. He is listed in Who's Who in Science and Engineering and Who's Who in Finance and Industry.

Li-Sun Shu (S'04) was born in Taiwan, R.O.C., in 1972. He received the B.S. degree in mathematics from Chung Yuan University, Chung-Li, Taiwan, R.O.C., in 1995 and the M.S. and Ph.D. degrees in information engineering and computer science from Feng Chia University, Taichung, Taiwan, R.O.C., in 1997 and 2004, respectively.

He is currently a Lecturer at the Overseas Chinese Institute of Technology. His research interests include evolutionary computation, large parameter optimization problems, fuzzy systems, and system optimization.