Hybrid Taguchi-Genetic Algorithm and Its Applications

周至宏 教授
(Prof. Jyh-Horng Chou)
國立高雄應用科技大學 電機工程系所
國立高雄第一科技大學 系統資訊與控制研究所

⊙演講內容係彙整自下列論文：

- J. H. Chou, S. H. Chen and J. J. Li, 2000, "Application of Taguchi-genetic method to design optimal grey-fuzzy controller of a constant turning force system", *J. of Materials Processing Technology*, Vol.105, pp.333-343. (also published in the *15th CSME Annual Conference*, Taiwan, November 1998.)

- J. T. Tsai, T. K. Liu and J. H. Chou, 2004, "Hybrid Taguchi-genetic algorithm for global numerical optimization", *IEEE Trans. on Evolutionary Computation*, Vol.8, pp.365-377. ***(ISI Highly Cited Paper)***

- J. T. Tsai, J. H. Chou and T. K. Liu, 2006, "Tuning the structure and parameters of a neural network by using hybrid Taguchi-genetic algorithm", *IEEE Trans. on Neural Networks*, Vol.17, pp.69-80.

- J. T. Tsai, J. H. Chou and T. K. Liu, 2006, "Optimal design of digital IIR filters by using hybrid Taguchi genetic algorithm", *IEEE Trans. on Industrial Electronics*, Vol.53, pp.867-879.

- J. T. Tsai, T. K. Liu, W. H. Ho and J. H. Chou, 2008, "An improved genetic algorithm for job-shop scheduling problems using Taguchi-based crossover", *Int. J. of Advanced Manufacturing Technology*, Vol.38, pp.987-994.

- W. H. Ho, J. T. Tsai and J. H. Chou, 2009, "Robust quadratic-optimal control of TS-fuzzy-model-based dynamic systems with both elemental parametric uncertainties and norm-bounded approximation error", *IEEE Trans. on Fuzzy Systems*, Vol.17, pp.518-531.

⊙中華民國發明專利(I220954)：

發明專利名稱：*Intelligent hybrid Taguchi-genetic algorithm*，專利期為:2004年9月至2021年2月。

⊙研究團隊之產學合作成果：

①協助尚富工業公司研發最佳自動給湯機構與控制設施(產學合作計畫總經費*二佰六十五萬元*)，使得尚富公司在設計自動給湯機構最佳化時，能有系統的從事設計工作及縮短設計時間。

②開發乙套自動給湯六軸多連桿機械手之機構設計和最佳運動路徑軟體，並以*二十二萬元*技轉給金屬工業研究發展中心。

③協助億尚精密工業股份有限公司(產學合作計畫總經費*六十四萬元*)進行平面顯示器玻璃鑽石輪刀切割機之多目標最佳化設計與研究。

④協助德冠企業公司(產學合作總經費*二十五萬元*)進行無線遠距智慧型動態即時生產排程監控暨擾動回覆系統研發。

Outline

- Genetic Algorithms
- Taguchi Method
- HTGA Approach
- Applications
- Conclusions

Genetic Algorithms

Genetic Algorithms

- Complete theory-structure and method for genetic algorithms have been developed by John Holland, his colleagues, and his students at the University of Michigan, Ann Arbor, 1975. (Original idea presented by A. S. Fraser in the Australian J. of Biological Science, Vol.10, pp.484-499, 1957.)
- Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics.
- Genetic algorithms have received considerable attention regarding their potential as a novel optimization technique for complex problems and have been successfully applied in various areas.

Pseudo Structure of Genetic Algorithms

```
Begin
    Initialization;
    Evaluation;
    While (not termination condition) do
        Selection;
        Crossover;
        Mutation;
        Evaluation;
    End
End
```

Representation

- Binary strings representation

1	1	1	0	0	1	1	0	1	1	0	1

- Real coding representation

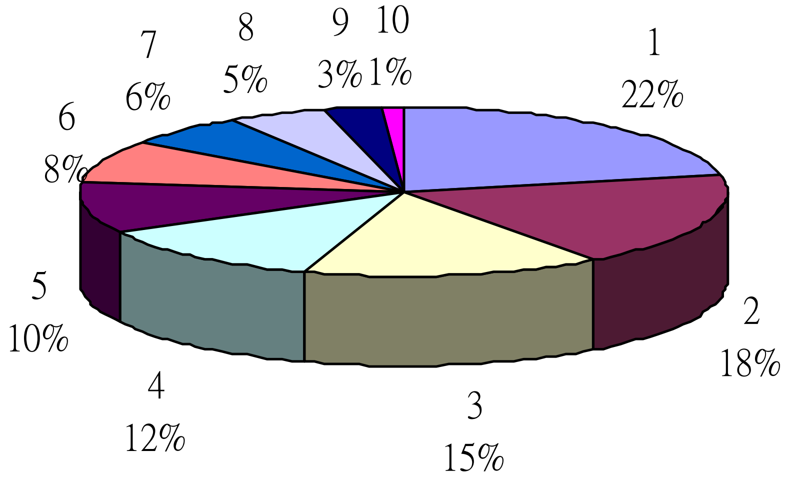
3.15	5.12	7.11	10.25	25.3

Representation (cont.)

- The real coding representation is accurate and efficient because it is closest to the real design space, and, moreover, the string length is the number of design variables. However, binary strings representing each variable with the desired precision are concatenated to represent an individual, the resulting string encoding a large number of design variables would wind up a huge string length.
- For example, for 100 variables with a precision of six digits, the string length is about 2000. The genetic algorithm would perform poorly for such design problems.

Selection

- The roulette wheel selection is employed.



- The basic idea is to determine selection probability for each chromosome proportional to the fitness value.

Crossover

- One-cut-point crossover

$$x=(x_1, x_2, \ldots, x_k, x_{k+1}, x_{k+2}, \ldots, x_n)$$
$$y=(y_1, y_2, \ldots, y_k, y_{k+1}, y_{k+2}, \ldots, y_n)$$

交叉點為 k 點

$$x'=(x_1, x_2, \ldots, x_k, y_{k+1}, y_{k+2}, \ldots, y_n)$$
$$y'=(y_1, y_2, \ldots, y_k, x_{k+1}, x_{k+2}, \ldots, x_n)$$

Crossover (cont.)

- Two-cut-point crossover

$$x=(x_1, x_2, x_3, \ldots, x_k, x_{k+1}, x_{k+2}, \ldots, x_n)$$

$$y=(y_1, y_2, y_3, \ldots, y_k, y_{k+1}, y_{k+2}, \ldots, y_n)$$

⇓ 交叉點為 2 與 k+2 點

$$x'=(x_1, x_2, y_3, \ldots, y_k, y_{k+1}, x_{k+2}, \ldots, y_n)$$

$$y'=(y_1, y_2, x_3, \ldots, x_k, x_{k+1}, y_{k+2}, \ldots, x_n)$$

Crossover (cont.)

- Uniform crossover

$$x=(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$$
$$y=(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$$

均匀交叉 01010101

$$x'= (x_1, y_2, x_3, y_4, x_5, y_6, x_7, y_8)$$
$$y'= (y_1, x_2, y_3, x_4, y_5, x_6, y_7, x_8)$$

Crossover (cont.)

- Arithmetical crossover (linear combination)

$$x=(x_1, x_2, \ldots, x_k, \ldots, x_n)$$
$$y=(y_1, y_2, \ldots, y_k, \ldots, y_n)$$

$$\Downarrow$$

$$x'=(x_1, x_2, \ldots, x'_k, \ldots, x_n)$$
$$y'=(y_1, y_2, \ldots, y'_k, \ldots, y_n)$$

- where $x'_k= \beta x_k+(1-\beta)\, y_k$, $y'_k= (1-\beta)\, x_k+\beta y_k$, and β is a random value, in which $\beta \in [0, 1]$.

Mutation

- Swap

$$x=(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$$

x_2 與 y_3 互換

$$x=(x_1, y_3, x_3, x_4, y_1, y_2, x_2, y_4)$$

Mutation (cont.)

- Backward insert

$$x=(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$$

x_2 向後插入 y_3 後

$$x=(x_1, x_3, x_4, y_1, y_2, y_3, x_2, y_4)$$

Mutation (cont.)

- Forward insert

$$x=(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$$

y_2 向前插入 x_1 後

$$x=(x_1, y_2, x_2, x_3, x_4, y_1, y_3, y_4)$$

Mutation (cont.)

- Arithmetical mutation (linear combination)

$$x=(x_1, x_2, \ldots, x_i, x_j, x_k, \ldots, x_n)$$

$$\Downarrow$$

$$x'=(x_1, x_2, \ldots, x'_i, x_j, x'_k, \ldots, x_n)$$

where $x'_i = (1-\beta)x_i + \beta x_k$, $x'_k = \beta x_i + (1-\beta)x_k$, and β is a random value, in which $\beta \in [0, 1]$.

Mutation (cont.)

- Arithmetical mutation (nonuniform mutation)

$$x=(x_1, x_2, \ldots, x_i, x_j, x_k, \ldots, x_n)$$

$$\Downarrow$$

$$x'=(x_1, x_2, \ldots, x'_i, x_j, x_k, \ldots, x_n)$$

where $x'_i = x_i + \Delta(t, x_i^U - x_i)$ or $x'_i = x_i - \Delta(t, x_i - x_i^L)$.

- The function $\Delta(t, y)$ returns a value in the range $[0, y]$ such that the value of $\Delta(t, y)$ approaches 0 as t increases (t is the generation number).
- $\Delta(t, y) = y \cdot \beta \cdot (1-t/T)^b$, where β is a random number from [0, 1], T is the maximal generation number, and b is a parameter determining the degree of nonuniformity.

Taguchi Method

Taguchi Method

- The Taguchi method barrows many ideas from the statistical design of experiments to evaluate and implement improvements into products, processes, or equipment.
- Its fundamental principle, largely speaking, is to improve the quality of concern by minimizing the effect of the causes of variation, but not eliminating the causes.
- Two major mathematical tools are used: (i) the signal-to-noise ratio, which is a measurement of the quality, and (ii) the orthogonal array, which reduces a large number of design parameters into usually a much smaller number of experiments.

Orthogonal Array

- For example, the general symbol for a two-level standard OA is

$$L_n(2^{n-1}),$$

where

$n = 2^k$ is the number of experimental runs and also is the number of rows,

k = a positive integer which is greater than 1,

2 = the number of levels for each factor considered, and

n-1 = the number of columns in the OA.

$L_8(2^7)$ Orthogonal Array

Experiment number	Factors						
	A	B	C	D	E	F	G
	Column number						
	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

$L_{16}(2^{15})$ Orthogonal Array

Experiment number	Column number														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
3	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2
4	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
5	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
6	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1
7	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1
8	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2
9	2	1	2	1	2	1	2	1	2	1	2	2	1	1	2
10	2	1	2	1	2	1	2	2	1	2	1	1	2	2	1
11	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1
12	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2
13	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1
14	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2
15	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2
16	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1

Orthogonal Array

- For example, the general symbol for a three-level standard OA is

$$L_m(3^{(m-1)/2}),$$

where

$m = 3^k$ is the number of experimental runs and also is the number of rows,

k = a positive integer which is greater than 1,

3 = the number of levels for each factor considered, and

$(m-1)/2$ = the number of columns in the OA.

$L_9(3^4)$ orthogonal array

Experiment number	Column number			
	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Signal-to-Noise Ratio

- A smaller-the-better characteristic:

$$\eta = -10 \log \left(\frac{1}{n}\sum_{t=1}^{n} y_t^2\right)$$

- A larger-the-better characteristic:

$$\eta = -10 \log \left(\frac{1}{n}\sum_{t=1}^{n} \frac{1}{y_t^2}\right)$$

HTGA Approach

Motivation

- A result is more robust if the relevant fitness values have a smaller standard deviation and, for this reason, the traditional genetic algorithm (TGA), which is largely based on the stochastic search techniques (Gen and Cheng, 1997), is less robust due to that the TGA usually has larger standard deviations.
- In this spirit, a natural question to ask is whether the Taguchi method may play a role in attaining more robust results, as the method has been well known to be a robust design approach.

Abstract

- The HTGA combines the traditional genetic algorithm (TGA), which has a powerful global exploration capability, with the Taguchi method, which can exploit the optimum offspring.
- The Taguchi method is inserted between crossover and mutation operations of a TGA. Then, the systematic reasoning ability of the Taguchi method is incorporated in the crossover operations to select the better genes to achieve crossover, and consequently enhance the genetic algorithm.

Introduction

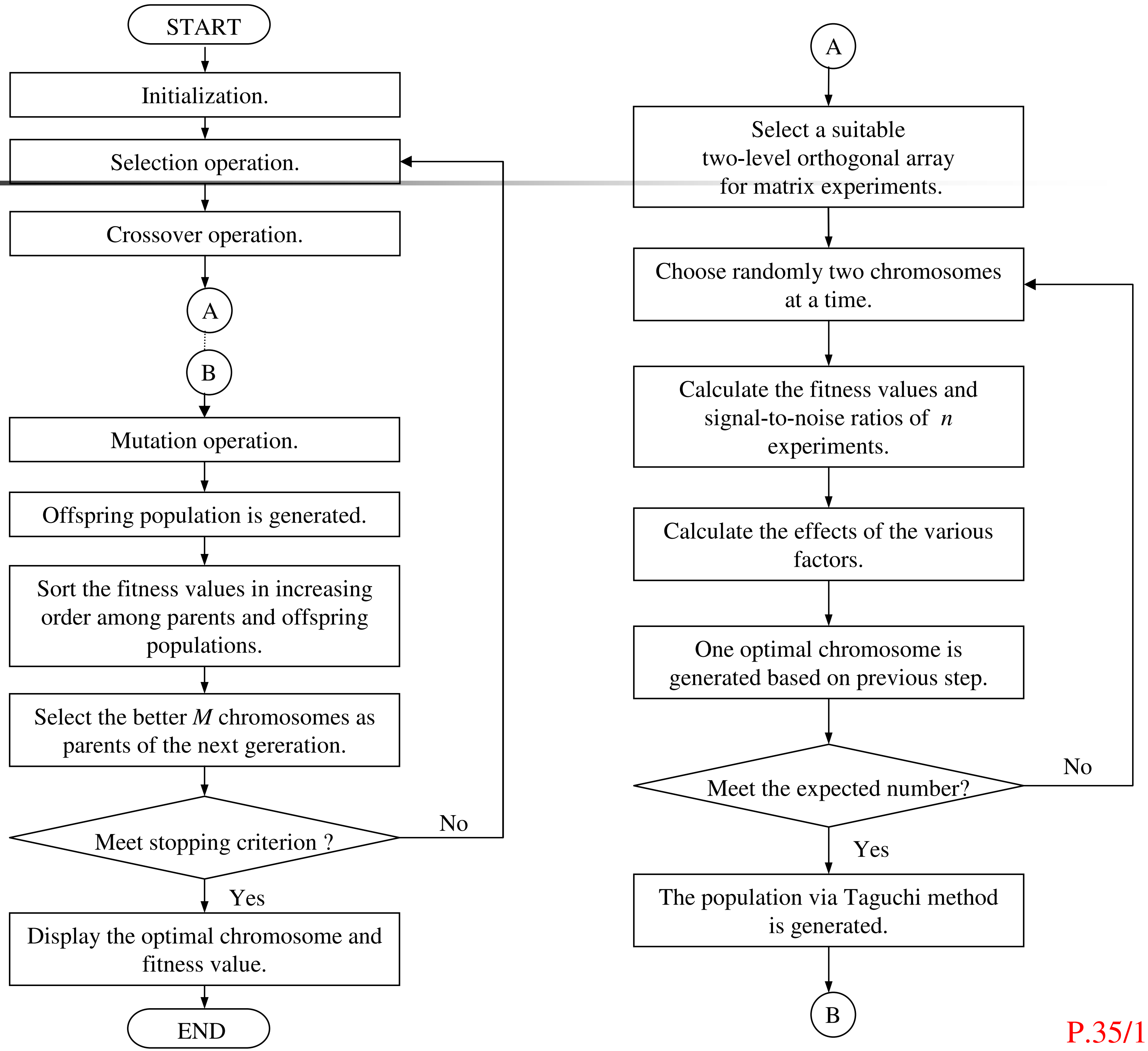
- Initially, improvements in the GA have been sought in the optimal proportion and adaptation of the main parameters, namely probability of mutation, probability of crossover, population size, and crossover operator (Grefenstette, 1986; Davis, 1989).

- More recently, attention has shifted to breeding (Angelov, 2001; Leung and Wang, 2001; Tsai et al., 2004).

Pseudo Structure of HTGA

```
Begin
    Initialization;
    Evaluation;
    While (not termination condition) do
        Selection;
        Crossover;
        Taguchi method;
        Mutation;
        Evaluation;
    End
End
```

Flow Chart of HTGA


START
Initialization.
Selection operation.
Crossover operation.
A
B
Mutation operation.
Offspring population is generated.
Sort the fitness values in increasing order among parents and offspring populations.
Select the better M chromosomes as parents of the next gereration.
Meet stopping criterion ?
No
Yes
Display the optimal chromosome and fitness value.
END
A
Select a suitable two-level orthogonal array for matrix experiments.
Choose randomly two chromosomes at a time.
Calculate the fitness values and signal-to-noise ratios of n experiments.
Calculate the effects of the various factors.
One optimal chromosome is generated based on previous step.
Meet the expected number?
No
Yes
The population via Taguchi method is generated.
B

Initialization

- The real coding technique is applied to solve optimization problems.

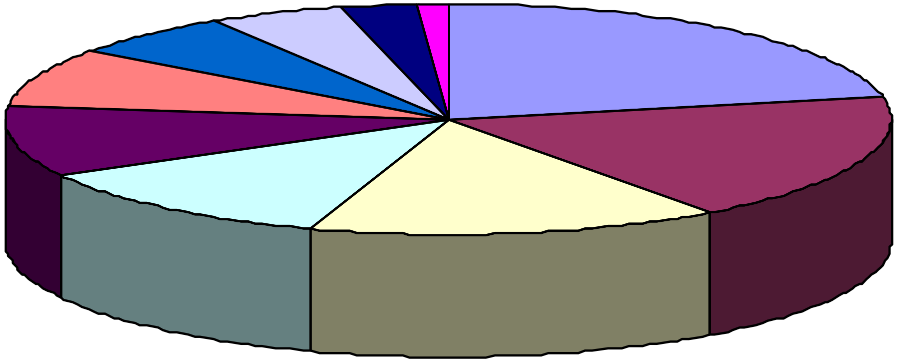
Step 1: Generate a random value β, where $\beta \in [0, 1]$.

Step 2: Let $x_i = l_i + \beta (u_i - l_i)$, where l_i and u_i are the domain of x_i. Repeat N times and produce a vector $(x_1, x_2, \ldots, x_i, \ldots, x_N)$.

Step 3: Repeat the above steps M times and produce M initial feasible solutions.

Selection Operation

- The roulette wheel selection is employed.



Crossover Operation

- The crossover operators used here are one-cut-point operator integrated with arithmetical operator derived from convex set theory (Bazaraa et al., 1990; Gen and Cheng, 1997), which randomly selects one cut-point, exchanges the right parts of two parents, and calculates the linear combinations at the cut-point genes to generate new offspring.

Crossover Operation (cont.)

$$x = (x_1, x_2, \ldots, x_k, x_{k+1}, x_{k+2}, \ldots, x_n)$$
$$y = (y_1, y_2, \ldots, y_k, y_{k+1}, y_{k+2}, \ldots, y_n)$$

$$\Downarrow$$

$$x' = (x_1, x_2, \ldots, x'_k, y_{k+1}, y_{k+2}, \ldots, y_n)$$
$$y' = (y_1, y_2, \ldots, y'_k, x_{k+1}, x_{k+2}, \ldots, x_n)$$

- where $x'_k = x_k + \beta(y_k - x_k)$, $y'_k = l_k + \beta(u_k - l_k)$, l_k and u_k are the domain of y_k, and β is a random value, in which $\beta \in [0, 1]$.

Taguchi Method

- The orthogonal arrays of the Taguchi method are used to study a large number of decision variables with a small number of experiments.
- The better combinations of decision variables are decided by the orthogonal arrays and the signal-to-noise ratios.
- A two-level orthogonal array is used.

Taguchi Method (cont.)

- The signal-to-noise ratio (η) refers to the mean-square-deviation of objective function.

- Let $\eta_i = y_i$ or $1/y_i$ if the objective function is to be maximized (larger-the-better) or minimized (smaller-the-better), respectively. Let y_i denote the fitness evaluation value of experiment i and i=1, 2, …, n, where n is experiment times.

Taguchi Method (cont.)

- The effects of the various factors (variables) can be defined as following:

$$E_{fl} = \text{sum of } \eta_i \text{ for factor } f \text{ at level } l$$

where i is the experiment number, f is the factor name, and l is the level number.

- For the two-level problem, if $E_{f1} > E_{f2}$, the optimum level is the level 1 for factor f. Otherwise, level 2 is the optimum one.

$L_8(2^7)$ Orthogonal Array

Experiment number	Factors						
	A	B	C	D	E	F	G
	Column number						
	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

- Chromosome c_1 are 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, and 0.0.
- Chromosome c_2 are 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, and 1.0.

Generating a better chromosome from two chromosomes by using the Taguchi method

Experiment Number (i)	Factors (f) A	B	C	D	E	F	G	Function value	η_i
	Column number 1	2	3	4	5	6	7		
1	1.0	1.0	1.0	1.0	0.0	0.0	0.0	4.0	1/4.0
2	1.0	1.0	1.0	0.0	1.0	1.0	1.0	6.0	1/6.0
3	1.0	0.0	0.0	1.0	0.0	1.0	1.0	4.0	1/4.0
4	1.0	0.0	0.0	0.0	1.0	0.0	0.0	2.0	1/2.0
5	0.0	1.0	0.0	1.0	1.0	0.0	1.0	4.0	1/4.0
6	0.0	1.0	0.0	0.0	0.0	1.0	0.0	2.0	1/2.0
7	0.0	0.0	1.0	1.0	1.0	1.0	0.0	4.0	1/4.0
8	0.0	0.0	1.0	0.0	0.0	0.0	1.0	2.0	1/2.0
E_{f1}	1.167*	1.167*	1.167	1	1.5	1.5	1.5		
E_{f2}	1.5*	1.5*	1.5	1.667	1.167	1.167	1.167		
Optimal level	2*	2*	2	2	1	1	1		
Optimal chromosome	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	κ**

Mutation Operation

- The basic concept of mutation operation is also derived from convex set theory.

$$x=(x_1,\ x_2,\ \ldots,\ x_i,\ x_j,\ x_k,\ldots,\ x_N)$$

$$\Downarrow$$

$$x'=(x_1,\ x_2,\ \ldots,\ x'_i,\ x_j,\ x'_k,\ldots,\ x_N)$$

where $x'_i= (1- \beta)x_i+\beta x_k$, $x'_k= \beta x_i+(1- \beta)x_k$, and β is a random value, in which $\beta \in [0, 1]$.

Application Example 1

- We adopt the well-known test functions in Table 1 to test our proposed HTGA, and to compare the performances of our proposed HTGA with the performances of OGA/Q presented by Leung and Wang (2001). (50 runs)

Table 1

Test function	Feasible solution space
$f_1 = \sum_{i=1}^{N} \left(-x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right).$	$[-500, 500]^N$
$f_2 = \sum_{i=1}^{N} \left(x_i^2 - 10\cos(2\pi x_i) + 10 \right).$	$[-5.12, 5.12]^N$
$f_3 = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right) + 20 + \exp(1).$	$[-32, 32]^N$
$f_4 = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$	$[-600, 600]^N$

Table 1 (cont.)

Test function	Feasible solution space
$$f_5 = \frac{\pi}{N}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{N-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_N - 1)^2\right\} + \sum_{i=1}^{N} u(x_i, 10, 100, 4),$$ where $$y_i = 1 + \frac{1}{4}(x_i + 1),$$ and $$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a. \\ 0, & -a \le x_i \le a. \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$$	$[-50, 50]^N$
$$f_6 = \frac{1}{10}\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{N-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_N - 1)^2\left[1 + \sin^2(2\pi x_N)\right]\right\} + \sum_{i=1}^{N} u(x_i, 5, 100, 4).$$	$[-50, 50]^N$

Table 1 (cont.)

Test function	Feasible solution space
$f_7 = -\sum_{i=1}^{N} \sin(x_i)\sin^{20}\left(\frac{i \times x_i^2}{\pi}\right).$	$[0,\ \pi]^N$
$f_8 = \sum_{i=1}^{N}\left[\sum_{j=1}^{N}(\chi_{ij}\sin\omega_j + \psi_{ij}\cos\omega_j) - \sum_{j=1}^{N}(\chi_{ij}\sin x_j + \psi_{ij}\cos x_j)\right]^2,$ where χ_{ij} and ψ_{ij} are random integers in $[-100, 100]$, and ω_{j} is a random number in $[-\pi, \pi]$.	$[-\pi,\ \pi]^N$
$f_9 = \frac{1}{N}\sum_{i=1}^{N}(x_i^4 - 16x_i^2 + 5x_i).$	$[-5, 5]^N$
$f_{10} = \sum_{j=1}^{N-1}\left[100(x_j^2 - x_{j+1})^2 + (x_j - 1)^2\right].$	$[-5, 10]^N$

Table 1 (cont.)

Test function	Feasible solution space
$f_{11} = \sum_{i=1}^{N} x_i^2$.	$[-100, 100]^N$
$f_{12} = \sum_{i=1}^{N} x_i^4 + \text{random}[0,1)$.	$[-1.28, 1.28]^N$
$f_{13} = \sum_{i=1}^{N} \lvert x_i \rvert + \prod_{i=1}^{N} \lvert x_i \rvert$.	$[-10, 10]^N$
$f_{14} = \sum_{i=1}^{N} \left(\sum_{j=1}^{i} x_j \right)^2$.	$[-100, 100]^N$
$f_{15} = \max\{ \lvert x_i \rvert, \quad i = 1,2,\ldots,N \}$.	$[-100, 100]^N$

Results and Comparisons

Test function	Mean number of function evaluations		Mean function value (standard deviation)		Globally minimal function value
	HTGA	OGA/Q	HTGA	OGA/Q	
f_1	163,468	302,166	-12569.4600 (0)	-12569.4537 (6.447×10^{-4})	-12569.5
f_2	16,267	224,710	0 (0)	0 (0)	0
f_3	16,632	112,421	0 (0)	4.440×10^{-16} (3.989×10^{-17})	0
f_4	20,999	134,000	0 (0)	0 (0)	0
f_5	66,457	134,556	1.000×10^{-6} (0)	6.019×10^{-6} (1.159×10^{-6})	0
f_6	59,003	134,143	1.000×10^{-4} (0)	1.869×10^{-4} (2.615×10^{-5})	0
f_7	265,693	302,773	-92.83 (0)	-92.83 (2.626×10^{-2})	-99.2784
f_8	186,816	190,031	5.869×10^{-5} (8.325×10^{-5})	4.672×10^{-7} (1.293×10^{-7})	0
f_9	216,535	245,930	-78.3030000 (0)	-78.3000296 (6.288×10^{-3})	-78.33236
f_{10}	60,737	167,863	7.000×10^{-1} (0)	7.520×10^{-1} (1.140×10^{-1})	0
f_{11}	20,844	112,559	0 (0)	0 (0)	0
f_{12}	20,065	112,652	1.000×10^{-3} (0)	6.301×10^{-3} (4.069×10^{-4})	0
f_{13}	14,285	112,612	0 (0)	0 (0)	0
f_{14}	26,469	112,576	0 (0)	0 (0)	0
f_{15}	21,261	112,893	0 (0)	0 (0)	0

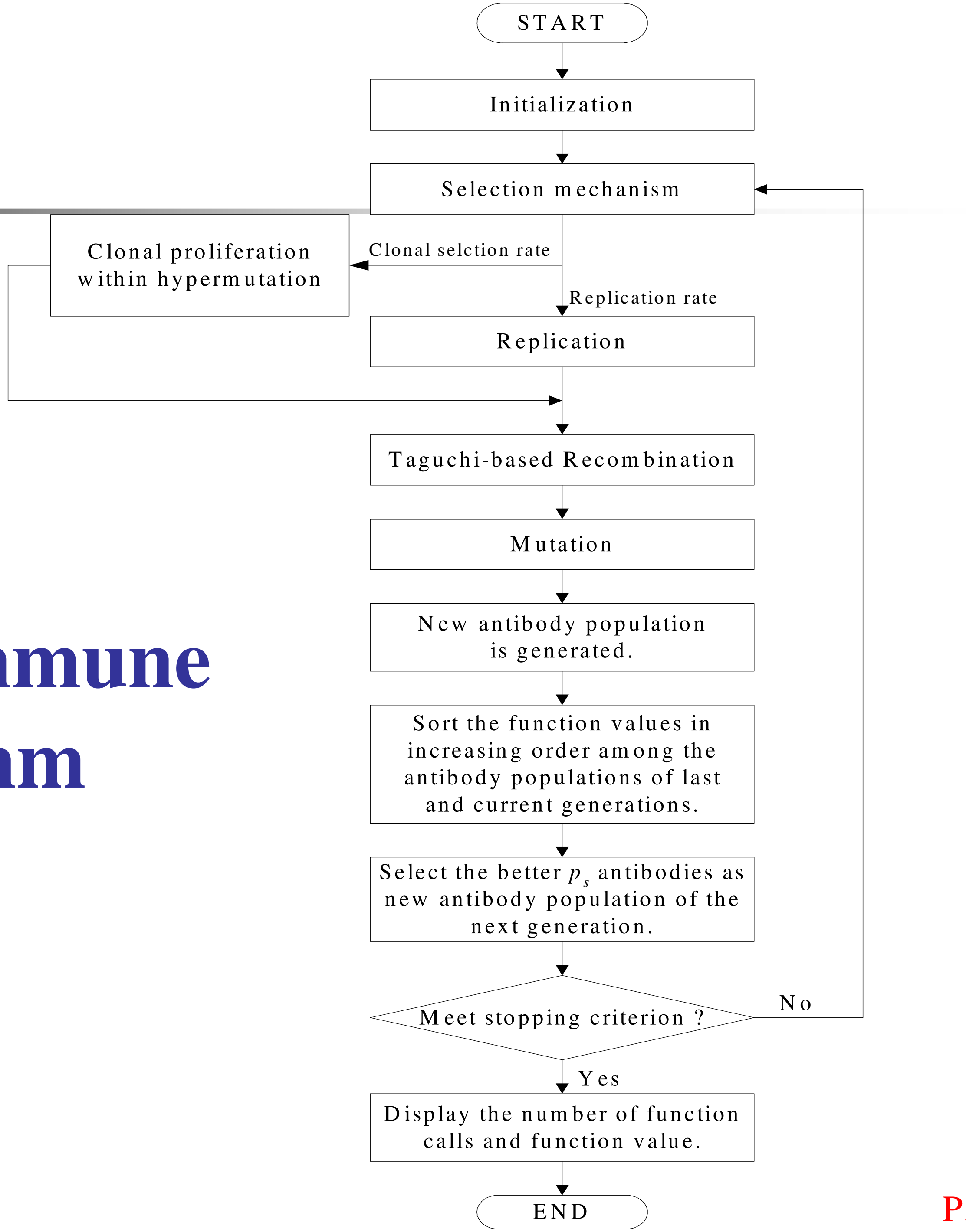
Results and Comparisons (cont.)

Test function	Mean number of function evaluations		Mean function value (standard deviation)		Globally minimal function value
	HTGA	OGA/Q	HTGA	OGA/Q	
f_1	110,923	302,166	-12569.4600 (0)	-12569.4537 (6.447×10^{-4})	-12569.5
f_2	15,105	224,710	0 (0)	0 (0)	0
f_3	14,122	112,421	0 (0)	4.440×10^{-16} (3.989×10^{-17})	0
f_4	16,085	134,000	0 (0)	0 (0)	0
f_5	34,274	134,556	1.000×10^{-6} (0)	6.019×10^{-6} (1.159×10^{-6})	0
f_6	20,582	134,143	1.000×10^{-4} (0)	1.869×10^{-4} (2.615×10^{-5})	0
f_7	219,006	302,773	-92.83 (0)	-92.83 (2.626×10^{-2})	-99.2784
f_8	185,483	190,031	1.697×10^{-5} (2.577×10^{-5})	4.672×10^{-7} (1.293×10^{-7})	0

Results and Comparisons (cont.)

Test function	Mean number of function evaluations		Mean function value (standard deviation)		Globally minimal function value
	HTGA	OGA/Q	HTGA	OGA/Q	
f_9	205,785	245,930	-78.3030000 (0)	-78.3000296 (6.288×10^{-3})	-78.33236
f_{10}	39,342	167,863	7.000×10^{-1} (0)	7.520×10^{-1} (1.140×10^{-1})	0
f_{11}	14,310	112,559	0 (0)	0 (0)	0
f_{12}	12,770	112,652	1.000×10^{-3} (0)	6.301×10^{-3} (4.069×10^{-4})	0
f_{13}	12,047	112,612	0 (0)	0 (0)	0
f_{14}	15,420	112,576	0 (0)	0 (0)	0
f_{15}	14,625	112,893	0 (0)	0 (0)	0

Taguchi-Immune Algorithm



Results and Comparisons (cont.)

Test function	Mean number of function evaluations			Mean function value (standard deviation)			Globally minimal function value
	TIA	HTGA	OGA/Q	TIA	HTGA	OGA/Q	
f_1	66,436	163,468	302,166	-12569.4600 (0)	-12569.4600 (0)	-12569.4537 (6.447×10^{-4})	-12569.5
f_2	13,267	16,267	224,710	0 (0)	0 (0)	0 (0)	0
f_3	14,996	16,632	112,421	0 (0)	0 (0)	4.440×10^{-16} (3.989×10^{-17})	0
f_4	15,743	20,999	134,000	0 (0)	0 (0)	0 (0)	0
f_5	23,491	66,457	134,556	1.000×10^{-6} (0)	1.000×10^{-6} (0)	6.019×10^{-6} (1.159×10^{-6})	0
f_6	21,837	59,003	134,143	1.000×10^{-4} (0)	1.000×10^{-4} (0)	1.869×10^{-4} (2.615×10^{-5})	0
f_7	197,024	265,693	302,773	-92.83 (0)	-92.83 (0)	-92.83 (2.626×10^{-2})	-99.2784
f_8	162,354	186,816	190,031	2.405×10^{-5} (4.410×10^{-5})	5.869×10^{-5} (8.325×10^{-5})	4.672×10^{-7} (1.293×10^{-7})	0
f_9	197,251	216,535	245,930	-78.3030000 (0)	-78.3030000 (0)	-78.3000296 (6.288×10^{-3})	-78.33236
f_{10}	19,067	60,737	167,863	7.000×10^{-1} (0)	7.000×10^{-1} (0)	7.520×10^{-1} (1.140×10^{-1})	0
f_{11}	15,623	20,844	112,559	0 (0)	0 (0)	0 (0)	0
f_{12}	11,475	20,065	112,652	1.000×10^{-3} (0)	1.000×10^{-3} (0)	6.301×10^{-3} (4.069×10^{-4})	0
f_{13}	12,864	14,285	112,612	0 (0)	0 (0)	0 (0)	0
f_{14}	20,831	26,469	112,576	0 (0)	0 (0)	0 (0)	0
f_{15}	14,431	21,261	112,893	0 (0)	0 (0)	0 (0)	0

Results and Comparisons (cont.)

Test function	Mean number of function evaluations			Mean function value (standard deviation)			Globally minimal function value
	TIA	HTGA	OGA/Q	TIA	HTGA	OGA/Q	
f_1	63,745	110,923	302,166	-12569.4600 (0)	-12569.4600 (0)	-12569.4537 (6.447×10^{-4})	-12569.5
f_2	13,059	15,105	224,710	0 (0)	0 (0)	0 (0)	0
f_3	13,964	14,122	112,421	0 (0)	0 (0)	4.440×10^{-16} (3.989×10^{-17})	0
f_4	13,424	16,085	134,000	0 (0)	0 (0)	0 (0)	0
f_5	15,415	34,274	134,556	1.000×10^{-6} (0)	1.000×10^{-6} (0)	6.019×10^{-6} (1.159×10^{-6})	0
f_6	13,624	20,582	134,143	1.000×10^{-4} (0)	1.000×10^{-4} (0)	1.869×10^{-4} (2.615×10^{-5})	0
f_7	177,348	219,006	302,773	-92.83 (0)	-92.83 (0)	-92.83 (2.626×10^{-2})	-99.2784
f_8	143,026	185,483	190,031	1.264×10^{-5} (1.742×10^{-5})	1.697×10^{-5} (2.577×10^{-5})	4.672×10^{-7} (1.293×10^{-7})	0
f_9	192,873	205,785	245,930	-78.3030000 (0)	-78.3030000 (0)	-78.3000296 (6.288×10^{-3})	-78.33236
f_{10}	16,463	39,342	167,863	7.000×10^{-1} (0)	7.000×10^{-1} (0)	7.520×10^{-1} (1.140×10^{-1})	0
f_{11}	13,055	14,310	112,559	0 (0)	0 (0)	0 (0)	0
f_{12}	9,080	12,770	112,652	1.000×10^{-3} (0)	1.000×10^{-3} (0)	6.301×10^{-3} (4.069×10^{-4})	0
f_{13}	11,507	12,047	112,612	0 (0)	0 (0)	0 (0)	0
f_{14}	12,023	15,420	112,576	0 (0)	0 (0)	0 (0)	0
f_{15}	11,435	14,625	112,893	0 (0)	0 (0)	0 (0)	0

Application Example 2

- Solve the problem of tuning both network structure and parameters of a feedforward neural network.

Problem Definition

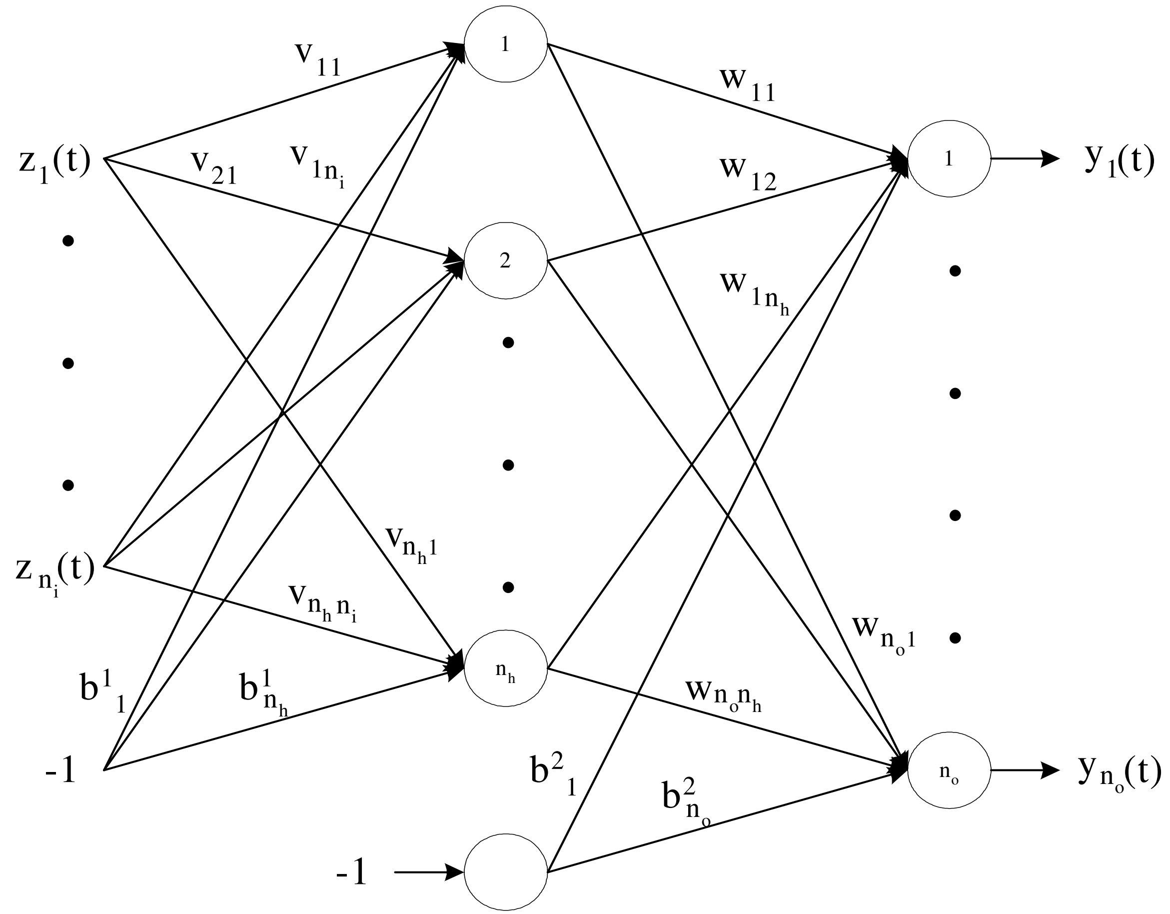
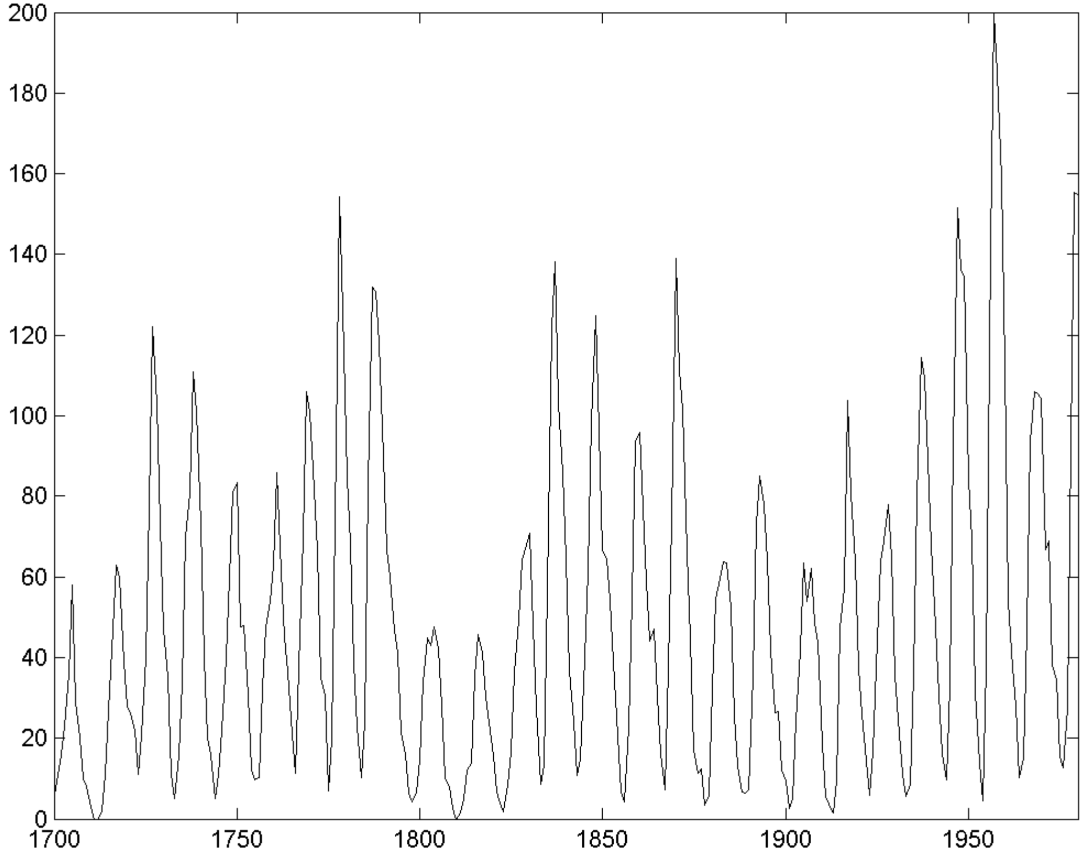


Figure 1. Three-layer feedforward neural network.

Sunspot Cycles

- Sunspot cycles from year 1700 to 1980.



Results and Comparisons

- Results of fitness values for the example on forecasting the sunspot numbers by using the HTGA approach. (20 runs)

n_h	Best	Average	Std. Dev.
4	0.95619	0.95580	0.00022
5	0.95627	0.95593	0.00028
6	0.95625	0.95608	0.00014
7	0.95622	0.95592	0.00025
8	0.95621	0.95566	0.00043

Results and Comparisons (cont.)

HTGA Approach				
n_h	Fitness Value	Training Error	Forecasting Error	Number of Links
4	0.95619	9.16351	13.81871	7
5	0.95627	9.14655	13.79772	8
6	0.95625	9.15012	13.87047	9
7	0.95622	9.15610	13.77482	10
8	0.95621	9.15985	13.73850	11

Method of Leung et al. (2001, 2003)				
n_h	Fitness Value	Training Error	Forecasting Error	Number of Links
4	0.9429	12.1116	13.9734	9
5	0.9448	11.6850	13.8354	17
6	0.9453	11.5730	14.0933	18
7	0.9426	12.1791	14.7434	13
8	0.9407	12.6076	14.3516	13

Application Example 3

- Solve the mixed H_2/H_∞ structure-specified controller design problems of multi-input and multi-output (MIMO) systems under both plant uncertainties and external disturbances.
- Consider the longitudinal control system of the supermaneuverable F18/HARV fighter aircraft (Voulgaris and Valavani, 1991; Chen and Cheng, 1998).

Problem Definition

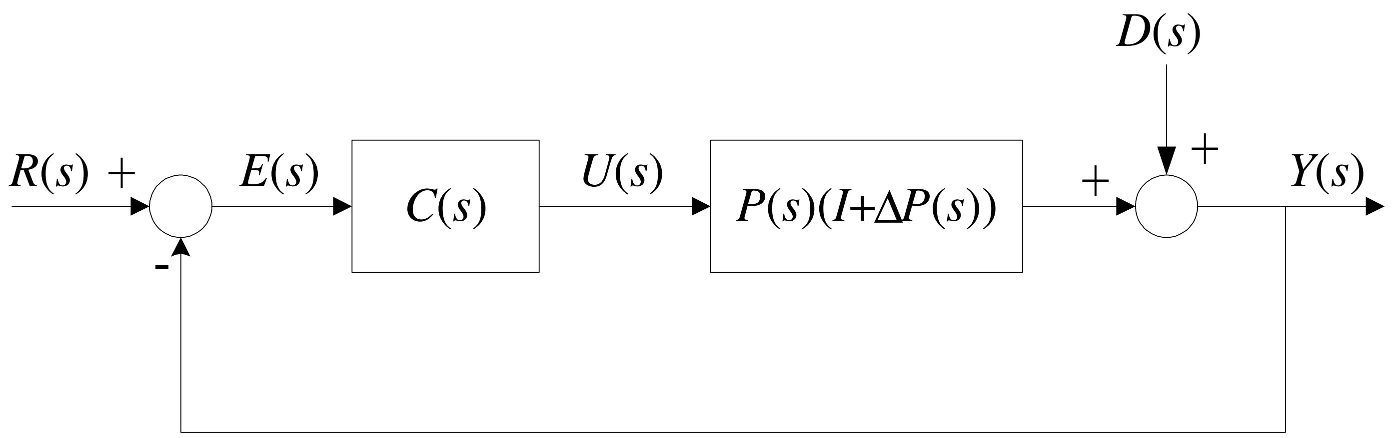


Figure 1. Control system with plant uncertainty and external disturbance.

Problem Definition (cont.)

- The structure-specified controller $C(s)$ is described by the following form:

$$C(s) = \frac{N_c(s)}{D_c(s)}$$

$$= \frac{Z_m s^m + Z_{m-1} s^{m-1} + \ldots + Z_0}{s^n + p_{n-1} s^{n-1} + \ldots + p_0},$$

with $$Z_k = \begin{bmatrix} z_{k11} & \cdots & z_{k1n_o} \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ z_{kn_i1} & \cdots & z_{kn_in_o} \end{bmatrix}$$

Problem Definition (cont.)

- By applying the H_∞ norm (Kwakernaak, 1985; Francis, 1987; Stoorvogel, 1992):

$$\kappa(\omega) = \sup_{\forall \omega \in [0, \infty)} \left((\bar{\sigma}(W_1(jw)T(jw)))^2 + (\bar{\sigma}(W_2(jw)S(jw)))^2 \right) < 1$$

- H_2 performance index:

$$J = \int_0^\infty (e^T(t)e(t))\, dt$$

Design Example

- The longitudinal dynamics of the system can be described with the following state space form (Voulgaris and Valavani, 1991):

$$\dot{x}(t) = Ax(t) + B_v u(t),$$
$$y(t) = Cx(t),$$

- External disturbance $d_1(t)=d_2(t)=d_3(t)=0.01e^{-0.2t}cos(3162.3t)$ and the plant uncertainty

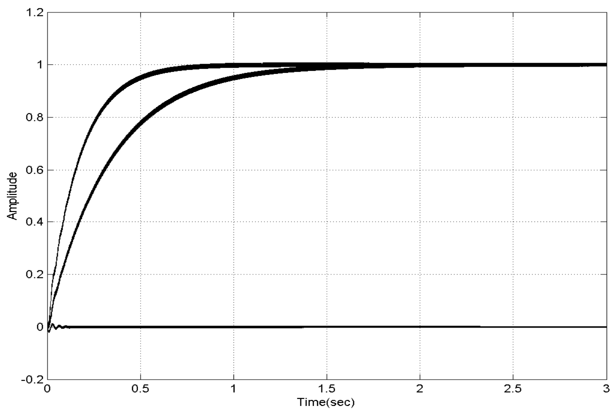
$$\Delta P(s) = \frac{0.012s^2 + 1.2s + 1.2}{s^2 + 20s + 100} I_{3\times 3}$$

- PI type controller

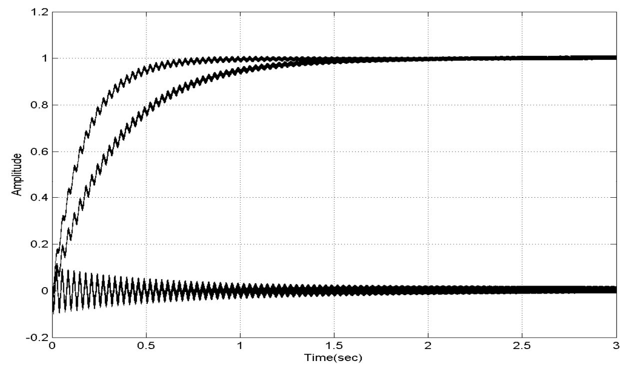
$$C(s) = \frac{\begin{bmatrix} z_{111} & z_{112} & z_{113} \\ z_{121} & z_{122} & z_{123} \\ z_{131} & z_{132} & z_{133} \end{bmatrix} s + \begin{bmatrix} z_{011} & z_{012} & z_{013} \\ z_{021} & z_{022} & z_{023} \\ z_{031} & z_{032} & z_{033} \end{bmatrix}}{s}$$

Results and Comparisons

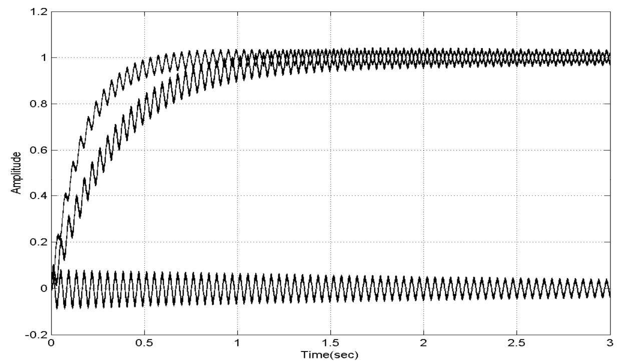
- Tracking responses for F18/HARV fighter aircraft control system that suffers from both plant uncertainties and external disturbances.



HTGA Approach



Approach of Ho et al.



Approach of Chen and Cheng

Application Example 4

- Solve the problem of designing the optimal digital infinite impulse response (IIR) filters.

Problem Statement

- Let $H(\omega, x)$ denote the transfer function of a digital filter, where x indicates the filter coefficients (e.g., poles and zeros). The magnitude of $H(\omega, x)$ is denoted as $|H(\omega, x)|$. The fundamental structure of $H(\omega, x)$, regardless of the filter type, can be given as

$$H(\omega, x) = \alpha\gamma\left(\prod_{i=1}^{M} \frac{1 + a_i e^{-j\omega}}{1 + b_i e^{-j\omega}}\right)\left(\prod_{k=1}^{N} \frac{1 + c_{1k} e^{-j\omega} + c_{2k} e^{-2j\omega}}{1 + d_{1k} e^{-j\omega} + d_{2k} e^{-2j\omega}}\right)$$

- where $x = (a_1, b_1, \ldots, a_M, b_M, c_{11}, c_{21}, d_{11}, d_{21}, \ldots, c_{1N}, c_{2N}, d_{1N}, d_{2N}, \gamma)$ denotes the filter parameters, and α is a positive normalizing constant such that the maximum value of $|H(\omega, x)|$ is 1.

Results and Comparisons

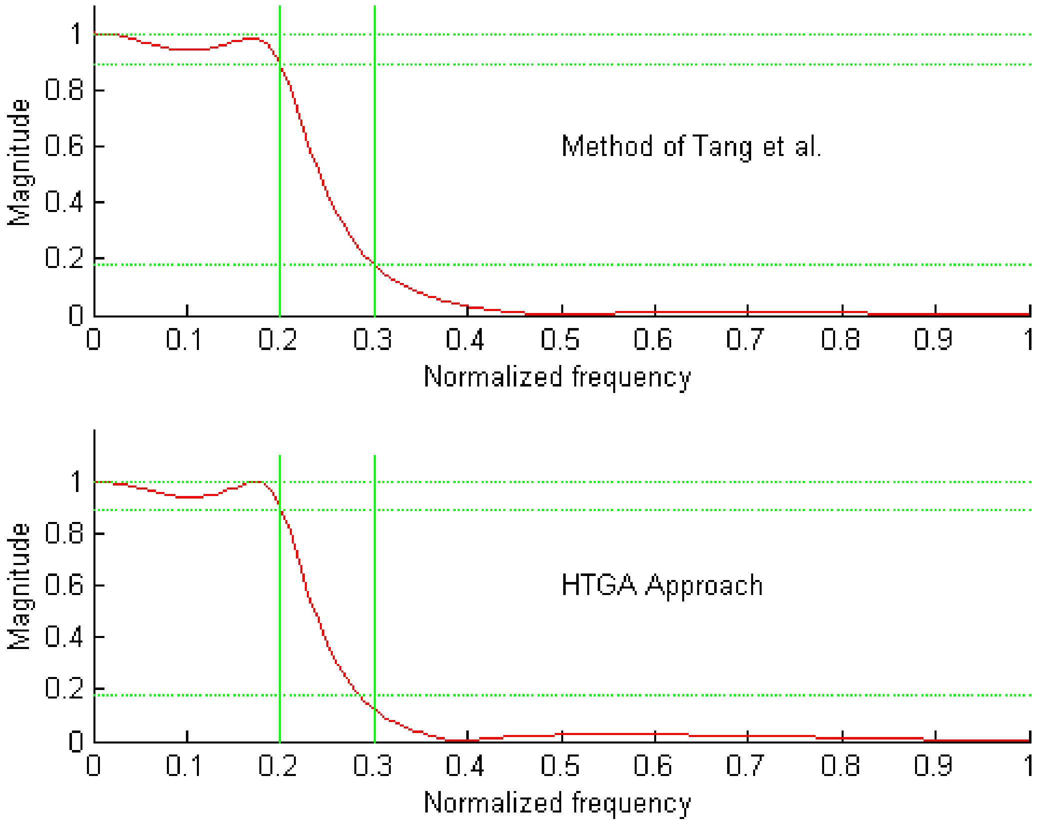
- We adopt the same examples and genetic operational parameters as those considered by Tang et al. (1998) to test our proposed HTGA approach, and to compare the performances of our proposed HTGA approach with those of the GA-based approach given by Tang et al. (1998).

Results and Comparisons (cont.)

HTGA Approach				
Filter type	L_1-norm error	L_2-norm error	Passband performance (Ripple magnitude)	Stopband performance (Ripple magnitude)
LP	4.2511	0.4213	$0.9004 \le \lvert H(e^{j\omega})\rvert \le 1.0000$ (0.0996)	$\lvert H(e^{j\omega})\rvert \le 0.1247$ (0.1247)
HP	4.8372	0.4558	$0.9460 \le \lvert H(e^{j\omega})\rvert \le 1.0000$ (0.0540)	$\lvert H(e^{j\omega})\rvert \le 0.1457$ (0.1457)
BP	1.9418	0.2350	$0.9760 \le \lvert H(e^{j\omega})\rvert \le 1.0000$ (0.0234)	$\lvert H(e^{j\omega})\rvert \le 0.0711$ (0.0711)
BS	4.5504	0.4824	$0.9563 \le \lvert H(e^{j\omega})\rvert \le 1.0000$ (0.0437)	$\lvert H(e^{j\omega})\rvert \le 0.1013$ (0.1013)
Method of Tang et al. (1998)				
Filter type	L_1-norm error	L_2-norm error	Passband performance (Ripple magnitude)	Stopband performance (Ripple magnitude)
LP	4.3395	0.5389	$0.8870 \le \lvert H(e^{j\omega})\rvert \le 1.0009$ (0.1139)	$\lvert H(e^{j\omega})\rvert \le 0.1802$ (0.1802)
HP	14.5078	1.2394	$0.9224 \le \lvert H(e^{j\omega})\rvert \le 1.0003$ (0.0779)	$\lvert H(e^{j\omega})\rvert \le 0.1819$ (0.1819)
BP	5.2165	0.6949	$0.8956 \le \lvert H(e^{j\omega})\rvert \le 1.0000$ (0.1044)	$\lvert H(e^{j\omega})\rvert \le 0.1772$ (0.1772)
BS	6.6072	0.7903	$0.8920 \le \lvert H(e^{j\omega})\rvert \le 1.0000$ (0.1080)	$\lvert H(e^{j\omega})\rvert \le 0.1726$ (0.1726)

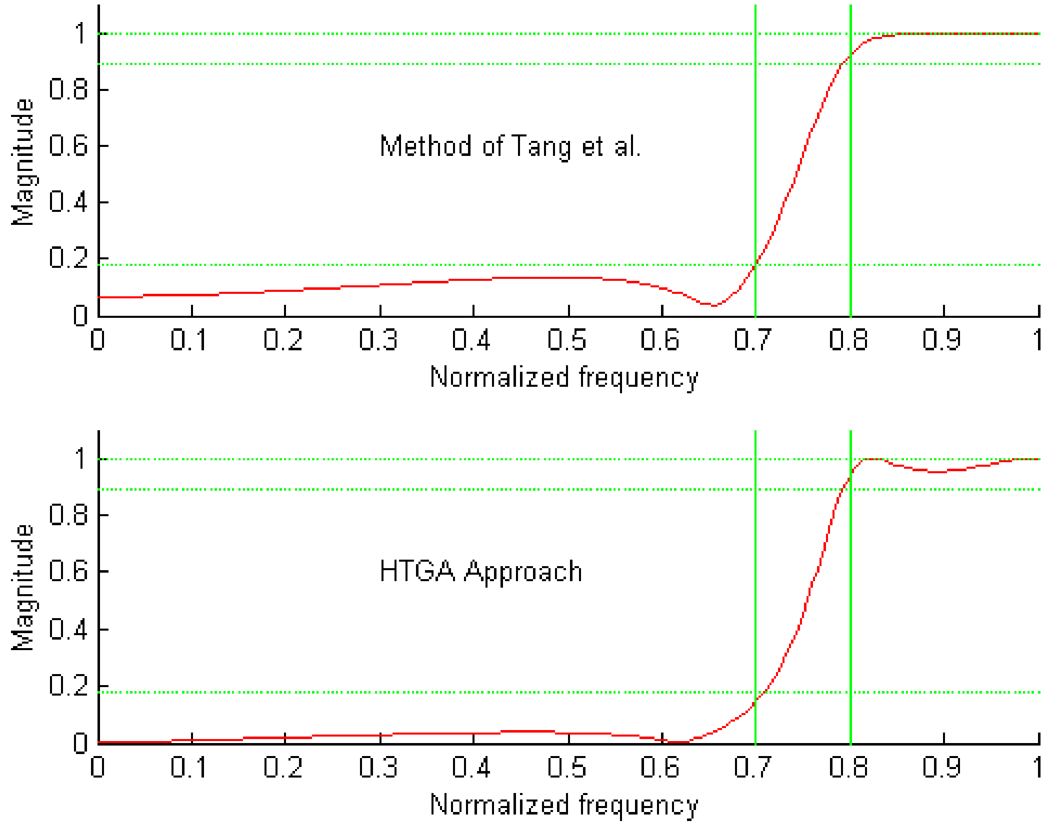
Results and Comparisons (cont.)

- Low-pass (LP) filter



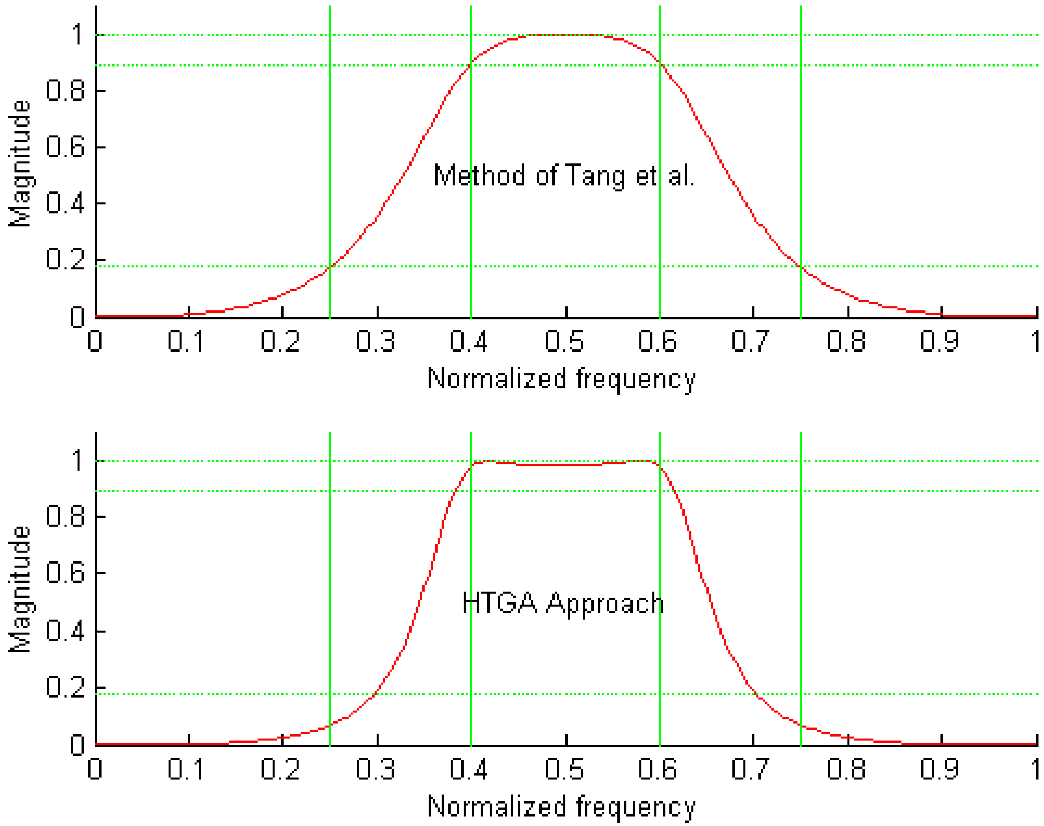
Results and Comparisons (cont.)

- High-pass (HP) filter



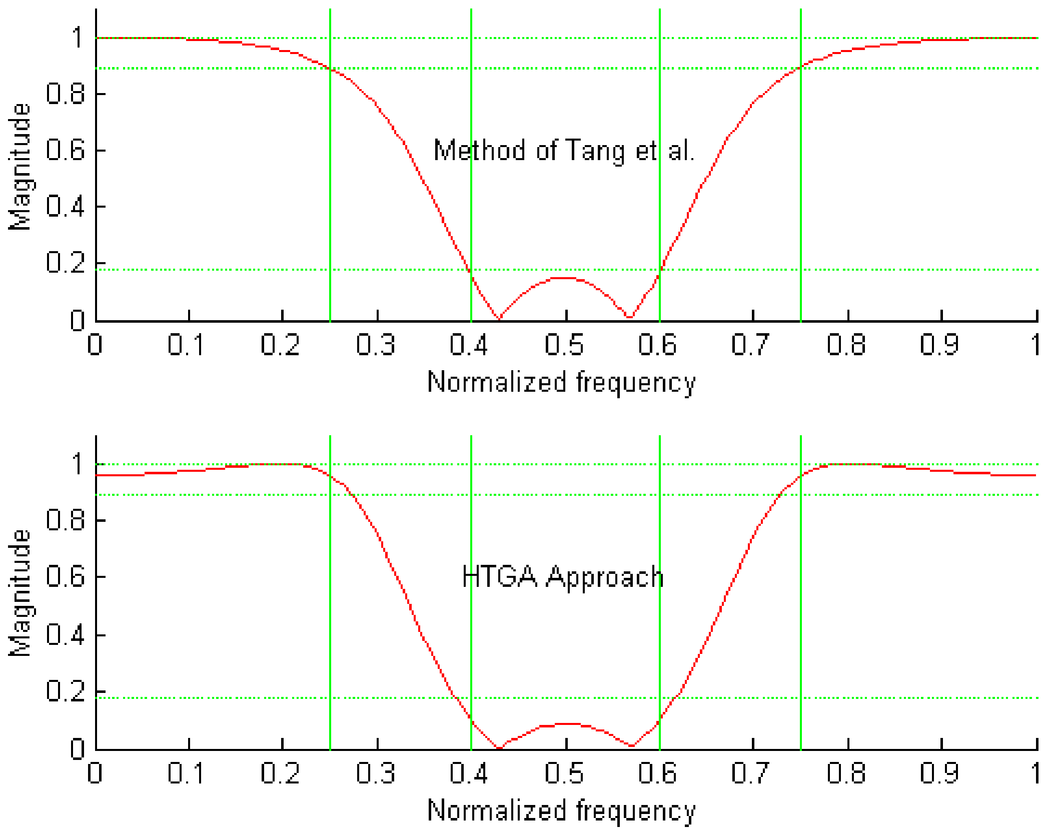
Results and Comparisons (cont.)

- Bandpass (BP) filter



Results and Comparisons (cont.)

- Bandstop (BS) filter



Application Example 5

- Solve the job-shop scheduling problem (JSP), which is one of the best-known machine scheduling problems and is among the hardest combinatorial optimization problems (Gen and Cheng, 1997).

Problem Statement

- The JSP is a scheduling problem that considers *M* different machines and *N* different jobs.
- Each of the jobs consists of *Q* operations and each of the operations requires a different machine.
- All the operations of each job are processed in the fixed processing order. Each operation is characterized by the required machine and the fixed processing time.
- A job is processed on one machine at a time and machines are available continuously.
- The JSP is a NP-hard problem, as the size of *M* machines and *N* jobs increases.

Schedule Encoding

- Job-shop scheduling is a sequencing problem of M machines and N jobs. Each job consists of a chain of operations. All the operations of each job are processed in the fixed processing order. Suppose that the available set of M machines is MC, and the index of the machine mc is k, then we have

$$MC = \{mc_k \mid k = 1,2,\ldots,M\}.$$

Schedule Encoding (cont.)

- Let the available set of total operations of N jobs be OP, and the indices of the operation op be i and k, then we have

  $$OP = \{op_{ik} \mid i = 1,2,\ldots,N \quad and \quad k = 1,2,\ldots,M\},$$

  where each entry in op_{ik} contains information such as job number, operation sequence of the job, machine number, and processing time of an operation. Thus, it can be formulated as a vector,

  $$op_{ik} = [job_i, operation_k, mc_k, t],$$

  where job_i is the number of a job, $operation_k$ indicates the operation sequence of job_i, mc_k denotes the machine to perform operation $operation_k$, and t is the processing time of operation $operation_k$ on machine mc_k.

Schedule Encoding (cont.)

- Consider a case of three jobs each of which has three operations. A nine-dimensional vector is constructed as:

  $$OP = (1,1,1,2,2,2,3,3,3) = (op_{11}, op_{12}, op_{13}, op_{21}, op_{22}, op_{23}, op_{31}, op_{32}, op_{33}),$$

  where, as an example, $op_{11} = [1,1,3,10], op_{12} = [1,2,2,8], op_{13} = [1,3,1,9],$ and so on. The first gene "1" of *OP* is the first operation of the job 1, which is performed on machine 3 and the processing time 10. The second gene "1" of *OP* is the second operation of job 1, which is performed on machine 2 and the processing time 8. The third gene "1" of *OP* is the third operation of job 1, which is performed on machine 1 and the processing time 9. The fourth gene "2" of *OP* is the first operation of job 2. The fifth gene "2" of *OP* is the second operation of job 2, and so on.

Taguchi-Based Crossover

- (Four-Job Four-Machine Example)

$U = \{1, 2, 3, 4\}$ $\Leftarrow$ the elements of 4-jobs set U

$(1\ 2\ 2\ 1)$ $\Leftarrow$ the values of the first 4 columns of the 3^{rd} experiment in $L_8(2^7)$

$\Downarrow$

$U_1 = \{1, 4\}$ and $U_2 = \{2, 3\}$

Chromosome A = $(\underline{1}, 2, 2, \underline{4}, 3, \underline{1}, 3, \underline{4}, 2, \underline{1}, 3, 3, 2, \underline{4}, \underline{1}, \underline{4})$

Chromosome B = $(\underline{2}, 1, \underline{2}, 4, \underline{3}, 1, 4, 4, \underline{3}, 1, \underline{2}, \underline{2}, 1, 4, \underline{3}, \underline{3})$

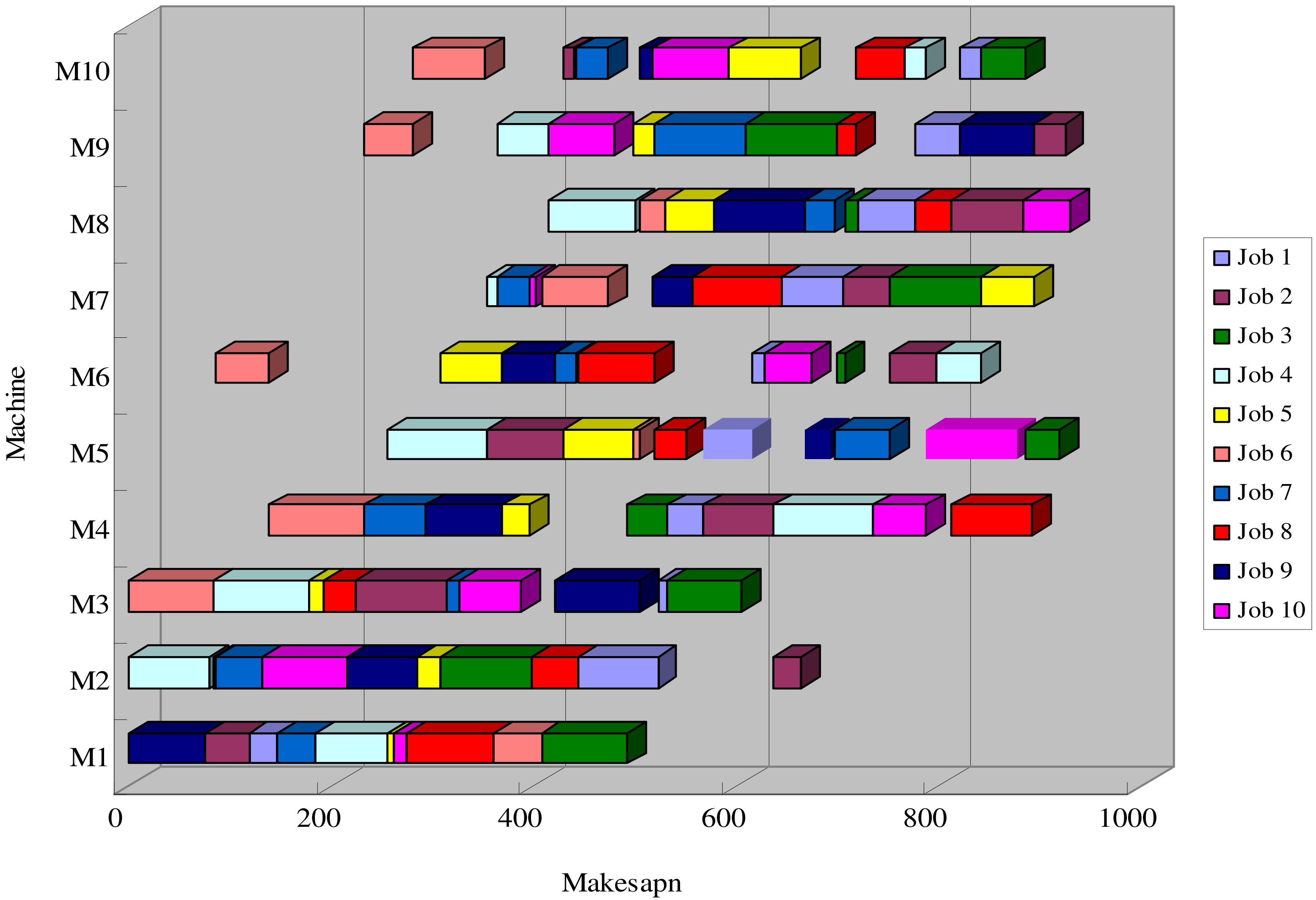
$\Downarrow$

New chromosome = (1, 2, 2, 4, 3, 1, 4, 3, 1, 2, 2, 4, 1, 3, 4, 3)

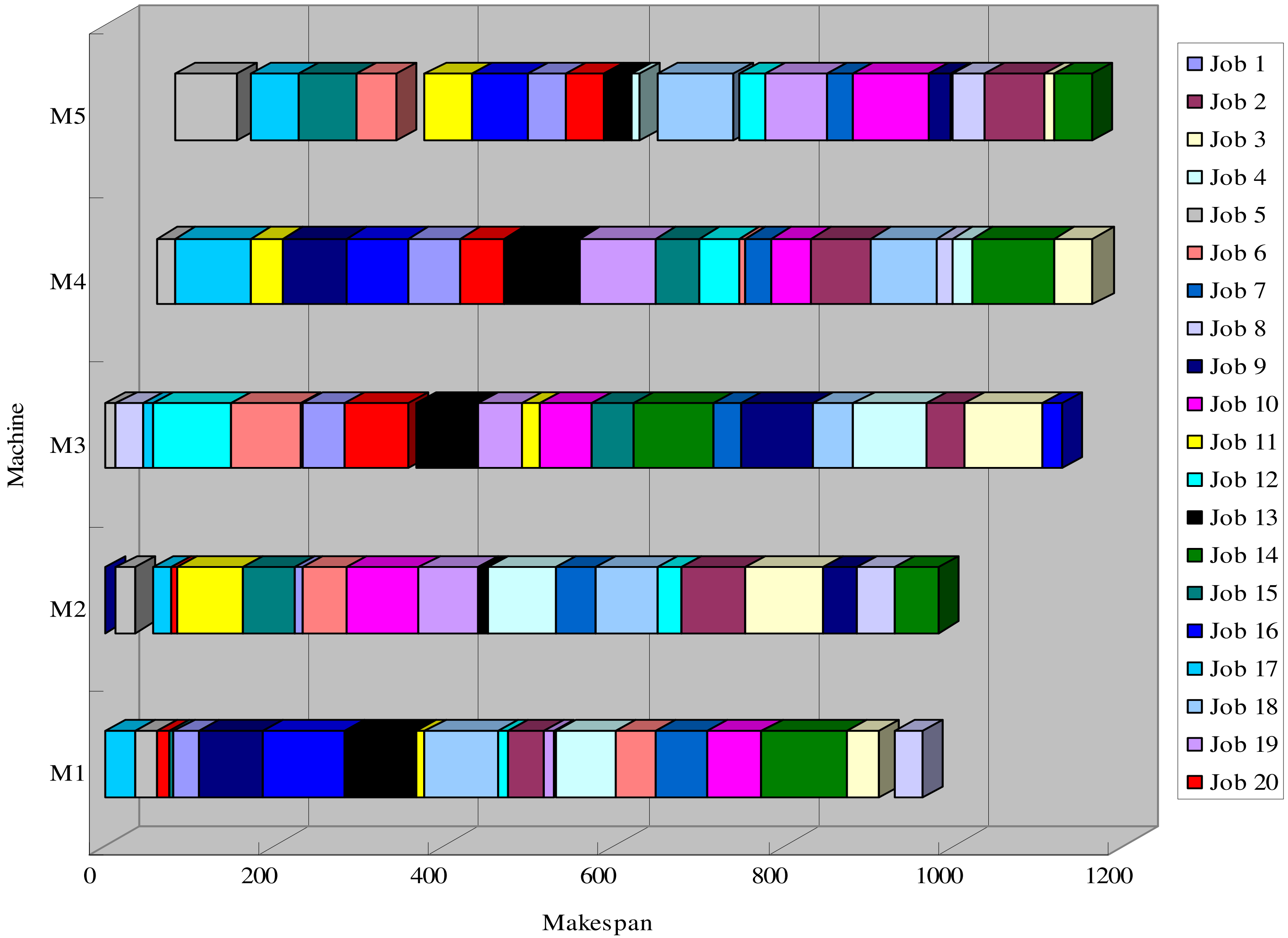
Results and Comparisons

- We adopt the same examples, the famous 10×10FT and 20×5FT problems, as those considered by Tsujimura et al. (2001) and Wang and Zheng (2001, 2002) to test our proposed HTGA approach, and to compare the performance of our proposed HTGA approach with the performances of those GA-based approaches given by Tsujimura et al. (2001) and Wang and Zheng (2001, 2002).

Results and Comparisons (cont.)



Results and Comparisons (cont.)



Results and Comparisons (cont.)

- Computational results in different methods for the 10×10FT (optimal: 930) (20 runs)

Method	GA	SyGA1	SyGA2	MGA	EVIS	HTGA	TBGA
Best	966	937	930	930	930	930	930
Average	993	965.85	965.5	953.7	930	943.48	930
Std. div.	19.45	15.85	17.09	NA	0	6.41	0

- GA, SyGA1, and SyGA2 are the methods proposed by Tsujimura et al. (2001).
- MGA is the method proposed by Wang and Zheng (2001, 2002).
- EVIS is the method proposed by Kim and Lee (1998).
- HTGA is the method proposed by Liu et al. (2006).

Results and Comparisons (cont.)

- Computational results in different methods for the 20×5FT (optimal: 1165) (20 runs)

Method	GA	SyGA1	SyGA2	MGA	EVIS	HTGA	TBGA
Best	1210	1189	1178	1165	1165	1165	1165
Average	1251.1	1214.9	1233.75	1179.22	1165.27	1172.5	1165
Std. div.	24.21	15.51	23.24	NA	NA	6.32	0

- GA, SyGA1, and SyGA2 are the methods proposed by Tsujimura et al. (2001).
- MGA is the method proposed by Wang and Zheng (2001, 2002).
- EVIS is the method proposed by Kim and Lee (1998).
- HTGA is the method proposed by Liu et al. (2006).

Results and Comparisons (cont.)

- Comparisons between MGA and TBGA for the Lawrence benchmarks

Problem	Size	Optimum	MGA			TBGA		
			Best	Average	Std. dev.	Best	Average	Std. dev.
LA11	20×5	1222	1222	1222	0	1222	1222	0
LA16	10×10	945	945	954.5	NA	945	945	0
LA21	15×10	1046	1058	1073.1	NA	1053	1056.5	1.96
LA26	20×10	1218	1218	1223.4	NA	1218	1218	0
LA31	30×10	1784	1784	1784	0	1784	1784	0

Results and Comparisons (cont.)

- Comparisons between EVIS and TBGA for the Lawrence benchmarks

Problem	Size	Optimum	EVIS			TBGA		
			Best	Average	Std. dev.	Best	Average	Std. dev.
LA11	20×5	1222	1222	1222	0	1222	1222	0
LA12	20×5	1039	1039	1039	0	1039	1039	0
LA13	20×5	1150	1150	1150	0	1150	1150	0
LA14	20×5	1292	1292	1292	0	1292	1292	0
LA15	20×5	1207	1207	1207	0	1207	1207	0
LA16	10×10	945	945	945.47	NA	945	945	0
LA17	10×10	784	784	784	0	784	784	0
LA18	10×10	848	848	848	0	848	848	0
LA19	10×10	842	842	848.23	NA	842	842	0
LA20	10×10	902	902	906.5	NA	902	906.5	1.58
LA21	15×10	1046	1055	1055.8	NA	1053	1056.5	1.96
LA22	15×10	927	935	935.47	NA	930	935	1.94
LA23	15×10	1032	1032	1032	0	1032	1032	0
LA24	15×10	935	941	959.37	NA	939	943.8	3.04
LA25	15×10	977	978	989.97	NA	984	984.5	1.58
LA26	20×10	1218	1218	1218	0	1218	1218	0
LA27	20×10	1235	1255	1264.9	NA	1246	1261.1	5.17
LA28	20×10	1216	1216	1224.63	NA	1216	1220.2	3.29
LA29	20×10	1152	1191	1196.6	NA	1167	1190.8	5.16
LA30	20×10	1355	1355	1355	0	1355	1355	0
LA31	30×10	1784	1784	1784	0	1784	1784	0
LA32	30×10	1850	1850	1850	0	1850	1850	0
LA33	30×10	1719	1719	1719	0	1719	1719	0
LA34	30×10	1721	1721	1721	0	1721	1721	0
LA35	30×10	1888	1888	1888	0	1888	1888	0

Application Example 6

- Robust Quadratic-Optimal Control of TS-Fuzzy-Model-Based Dynamic Systems with Both Elemental Parametric Uncertainties and Norm-Bounded Approximation Error

Problem Statement

Consider the following nonlinear control system with parametric uncertainties:

$$\dot{x}(t) = f(x(t), \theta(t)) + g(x(t), \theta(t))u(t). \quad (1)$$

With N the number of rules, the nonlinear control system with parametric uncertainties in (1) can be represented by the following TS(Takagi-Sugeno)-fuzzy-model-based dynamic system with both parametric uncertainties and approximation error:

$$\dot{x}(t) = \sum_{i=1}^{N} h_i(z(t))\left[(A_i + \Delta A_i(t))x(t) + (B_i + \Delta B_i(t))u(t)\right] + \Delta f + \Delta\tilde{g}u(t), \quad (2)$$

where $$\Delta A_i(t) = \sum_{k=1}^{a} \varepsilon_{ik}(t)E_{ik} \quad \text{and} \quad \Delta B_i(t) = \sum_{k=1}^{a} \eta_{ik}(t)V_{ik}. \quad (3)$$

Problem Statement (cont.)

Here, we consider the following parallel-distributed-compensation (PDC) controller:

$$u(t) = -\sum_{i=1}^{N} h_i(z(t)) F_i x(t), \tag{4}$$

By substituting (3) and (4) into (2), we can get the closed-loop uncertain TS-fuzzy-model-based dynamic system as

$$\dot{x}(t) = \sum_{i=1}^{N} \sum_{j=1}^{N} h_i(z(t)) h_j(z(t)) \left(A_i + \sum_{k=1}^{a} \varepsilon_{ik}(t) E_{ik} - \left(B_i + \sum_{k=1}^{a} \eta_{ik}(t) V_{ik} \right) F_j \right) x(t) + \Delta f + \Delta g, \tag{5}$$

where

$$\Delta g = -\sum_{j=1}^{N} h_j(z(t)) \Delta \tilde{g} F_j x(t). \tag{6}$$

Problem Statement (cont.)

As the works of Chen et al. (1999), it is assumed that there exist bounding matrices $\Delta\tilde{A}_i$ and $\Delta\tilde{B}_i$ such that

$$\left\|\Delta f\right\| \le \left\|\sum_{i=1}^{N} h_i\left(z(t)\right)\Delta\tilde{A}_i x(t)\right\| = \left\|\sum_{i=1}^{N} h_i\left(z(t)\right)\delta_i A_p x(t)\right\| \le \left\|A_p x(t)\right\| \tag{7}$$

and

$$\left\|\Delta g\right\| \le \left\|\sum_{i=1}^{N}\sum_{j=1}^{N} h_i\left(z(t)\right)h_j\left(z(t)\right)\Delta\tilde{B}_i F_j x(t)\right\| = \left\|\sum_{i=1}^{N}\sum_{j=1}^{N} h_i\left(z(t)\right)h_j\left(z(t)\right)\xi_i B_p F_j x(t)\right\| \le \left\|\sum_{j=1}^{N} h_j\left(z(t)\right)B_p F_j x(t)\right\|. \tag{8}$$

Problem Statement (cont.)

Theorem:

For the specified local feedback gain matrices F_i in (4), there exist a symmetric positive definite matrix P such that the following linear-matrix-inequalities (LMIs) are simultaneously satisfied:

$$\begin{bmatrix} U_{ijl}^{\mathrm{T}} P + P U_{ijl} + 2I & \left(PB_p F_j\right)^{\mathrm{T}} & P \\ PB_p F_j & -I & 0 \\ P & 0 & -\left(A_p^{\mathrm{T}} A_p\right)^{-1} \end{bmatrix} < 0, \tag{9}$$

for $i, j = 1, 2, \ldots, N$, and $l = 1, 2, \ldots, 2^{2a}$, where I denotes the identity matrix, and

$$U_{ijl} = \sum_{k=0}^{a} \left(\varepsilon_{ik}(t) E_{ik}^{j} - \eta_{ik}(t) L_{ijk}\right)\Bigg|_{\substack{\varepsilon_{ik}(t) = \underline{\varepsilon}_{ik} \text{ or } \bar{\varepsilon}_{ik} \\ \eta_{ik}(t) = \underline{\eta}_{ik} \text{ or } \bar{\eta}_{ik}}} . \tag{10}$$

Problem Statement (cont.)

The problem considered is how to specify the local feedback gain matrices F_i $(i = 1, 2, \ldots, N)$ of the PDC controller (4) such that (i) the constraint of LMI-based robust stabilizability condition (11) can be satisfied, and (ii) the optimal control performance for the nominal TS-fuzzy-model-based dynamic system

$$\dot{x}(t) = \sum_{i=1}^{N} h_i(z(t))\left(A_i x(t) + B_i u(t)\right) \tag{11}$$

can be achieved by minimizing the following H_2 quadratic finite-horizon integral performance index:

$$J = \int_0^{q t_f} \left[x^{\mathrm{T}}(t) Q x(t) + u^{\mathrm{T}}(t) R u(t)\right] dt = \sum_{k=0}^{q-1} \int_{k t_f}^{(k+1) t_f} \left[x^{\mathrm{T}}(t) Q x(t) + u^{\mathrm{T}}(t) R u(t)\right] dt. \tag{12}$$

Problem Statement (cont.)

Use a systematic approach for specifying the local feedback gain matrices F_i $(i = 1, 2, \ldots, N)$ of the PDC controller (4) to do the following two steps:

Step 1: check the constraint of LMI-based robust stabilizability condition (9),

Step 2: minimize the H_2 quadratic finite-horizon integral performance index (12) for the nominal TS-fuzzy-model-based dynamic system (11).

Robust Quadratic-Optimal PDC Controllers Design

The state vector $x(t)$ can be represented by the truncated orthogonal functions (OF) as

$$x(t) = \sum_{s=0}^{m-1} x_s^{(k)} T_s(t) = \tilde{x}^{(k)} T(t), \tag{13}$$

The quadratic integral performance index J becomes the following algebraic form:

$$J = \sum_{k=0}^{q-1} \text{trace}\left[W\left(\tilde{x}^{(k)}\right)^{\mathrm{T}} \left(Q + \sum_{i=1}^{N}\sum_{j=1}^{N} h_i(z_k) h_j(z_k) F_i^{\mathrm{T}} R F_j \right) \left(\tilde{x}^{(k)}\right) \right]. \tag{14}$$

The coefficient matrix $\tilde{x}^{(k)}$ can be obtained as

$$\hat{x}^{(k)} = \left[I_{mn} - \sum_{i=1}^{N}\sum_{j=1}^{N} h_i(z_k) h_j(z_k) \left(H^{\mathrm{T}} \otimes \left(A_i - B_i F_j \right) \right) \right]^{-1} \hat{Q}^{(k)}. \tag{15}$$

Robust Quadratic-Optimal PDC Controllers Design (cont.)

The value of the performance index of algebraic form in (14) is actually dependent on the set of local feedback gain matrices $\{F_1, F_2, \ldots, F_N\}$, which means

$$J = G(f_{111}, f_{112}, \ldots, f_{Nrn}), \tag{16}$$

The design problem of the robust quadratic-optimal PDC controller is to search for the optimal f_{ijk} such that (i) there exist a symmetric positive definite matrix P to make the LMIs in (9) hold, and (ii) the performance index of algebraic form in (14) is minimized. This is equivalent to the static constrained-optimization problem

$$\min \quad J = G(f_{111}, f_{112}, \ldots, f_{Nrn}) \tag{17}$$

The HTGA can be employed to search for the optimal solution of (17).

Illustrative Example

A balancing problem of an inverted pendulum on a cart is considered and given below:

$$\dot{x}_1(t) = x_2(t), \tag{18a}$$

$$\dot{x}_2(t) = \frac{1}{(\bar{M}+\bar{m})(\bar{J}+\bar{m}\bar{l}^2) - (\bar{m}\bar{l}\cos x_1(t))^2}$$

$$\cdot \Big[-\bar{f}_0(\bar{M}+\bar{m})x_2(t) - (\bar{m}\bar{l}x_2(t))^2 \sin x_1(t) \cos x_1(t)$$

$$+ (\bar{M}+\bar{m})\bar{m}\bar{g}\bar{l}\sin x_1(t) - \bar{m}\bar{l}\cos x_1(t) u(t) \Big], \tag{18b}$$

Illustrative Example (cont.)

We approximate the system with the nominal friction factor by the following four-rules TS fuzzy model:

$$\tilde{R}^1: \text{ IF } x_1(t) \text{ is about } 0 \text{ rad, THEN } \dot{x}(t) = A_1 x(t) + B_1 u(t), \tag{19a}$$

$$\tilde{R}^2: \text{ IF } x_1(t) \text{ is about } \pm\pi/9 \text{ rad, THEN } \dot{x}(t) = A_2 x(t) + B_2 u(t), \tag{19b}$$

$$\tilde{R}^3: \text{ IF } x_1(t) \text{ is about } \pm 2\pi/9 \text{ rad, THEN } \dot{x}(t) = A_3 x(t) + B_3 u(t), \tag{19c}$$

$$\tilde{R}^4: \text{ IF } x_1(t) \text{ is about } \pm\pi/3 \text{ rad, THEN } \dot{x}(t) = A_4 x(t) + B_4 u(t), \tag{19d}$$

Illustrative Example (cont.)

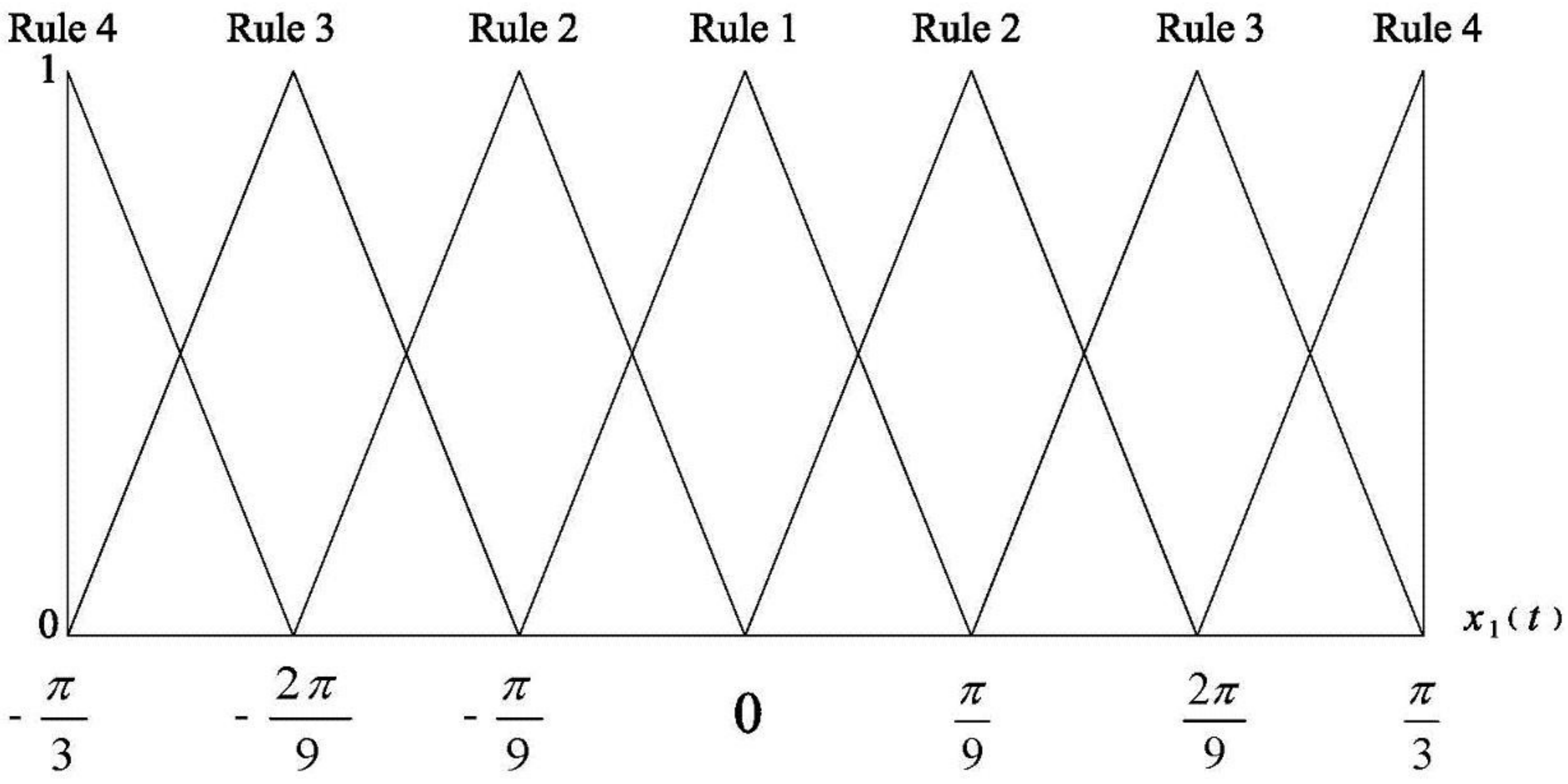


Fig. 1. Membership functions of the TS-fuzzy-model-based system .

Illustrative Example (cont.)

It is assumed that the parametric uncertainty of this example arises from the uncertain friction factor $\bar{f}(t)$. The TS fuzzy model with both one elemental parametric uncertainty and norm-bounded approximation error is of the form

$$\dot{x}(t)=\sum_{i=1}^{4} h_i(x_1(t))[(A_i+\Delta A_i(t))x(t)+B_i u(t)]+\Delta f+\Delta g, \tag{20}$$

The quadratic finite-horizon integral performance index is

$$J=\int_0^5\left[x^{\mathrm{T}}(t)Qx(t)+u^{\mathrm{T}}(t)Ru(t)\right]dt=\sum_{k=0}^{q-1}\int_{k t_f}^{(k+1)t_f}\left[x^{\mathrm{T}}(t)Qx(t)+u^{\mathrm{T}}(t)Ru(t)\right]dt, \tag{21}$$

Illustrative Example (cont.)

After using the proposed integrative approach and the LMI toolbox, the optimal local feedback gain matrices are

$$F_1 = [-237.1627 \quad -37.1691], \quad F_2 = [-365.3540 \quad -91.0221],$$

$$F_3 = [-263.4661 \quad -68.1953], \quad F_4 = [-241.5067 \quad -50.1357],$$

and a symmetric positive definite matrix P is

$$P = \begin{bmatrix} 0.8193 & 0.1738 \\ 0.1738 & 0.0496 \end{bmatrix}.$$

And we have the bounding matrices $A_p = \begin{bmatrix} 0 & 0.0080 \\ 0.17216 & 0.00224 \end{bmatrix}$ and $B_p = \begin{bmatrix} 0 \\ 0.00546 \end{bmatrix}$.

The inequalities, $\|\Delta f\| \le \|A_p x(t)\|$ and $\|\Delta g\| \le \left\| \sum_{j=1}^{4} h_j(z(t)) B_p F_j x(t) \right\|$, are satisfied.

Illustrative Example (cont.)

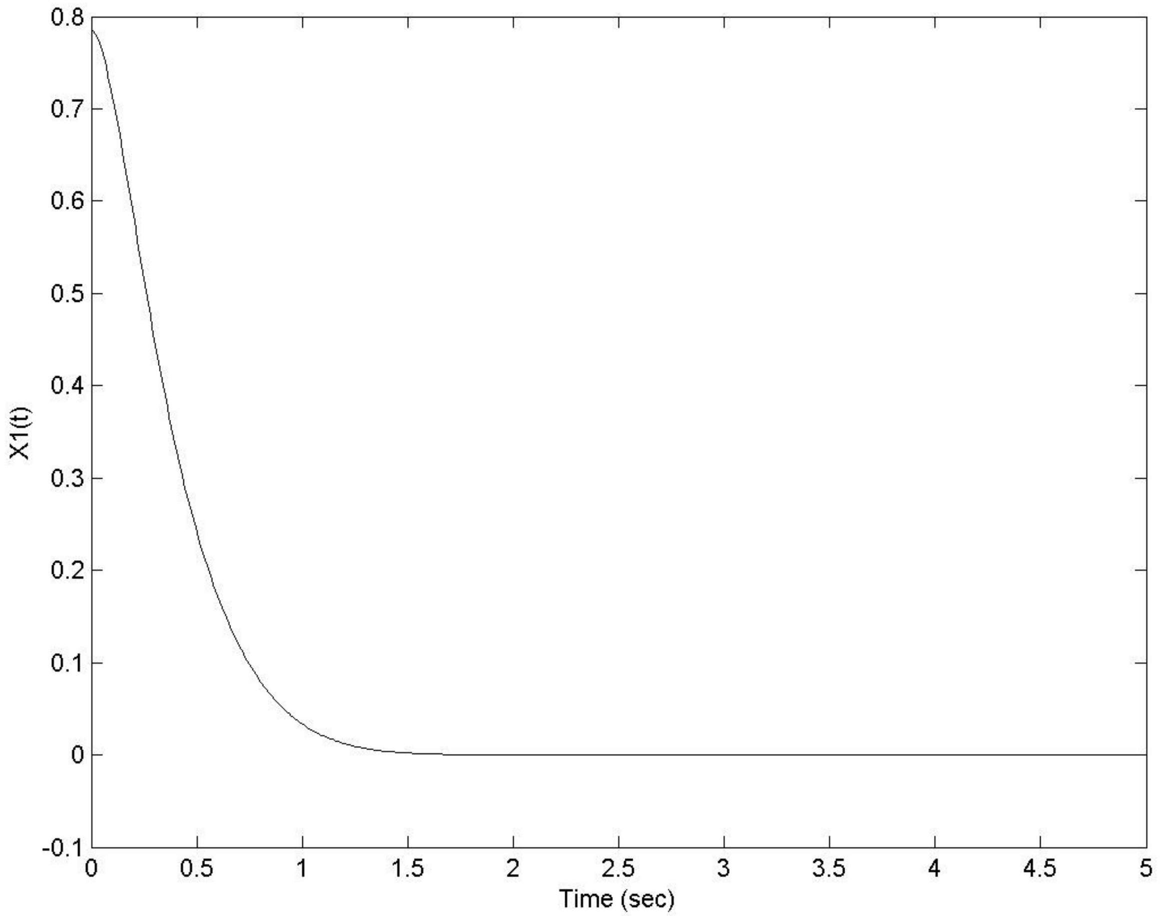


Fig. 2. Response of the angle $x_1(t)$.

Illustrative Example (cont.)

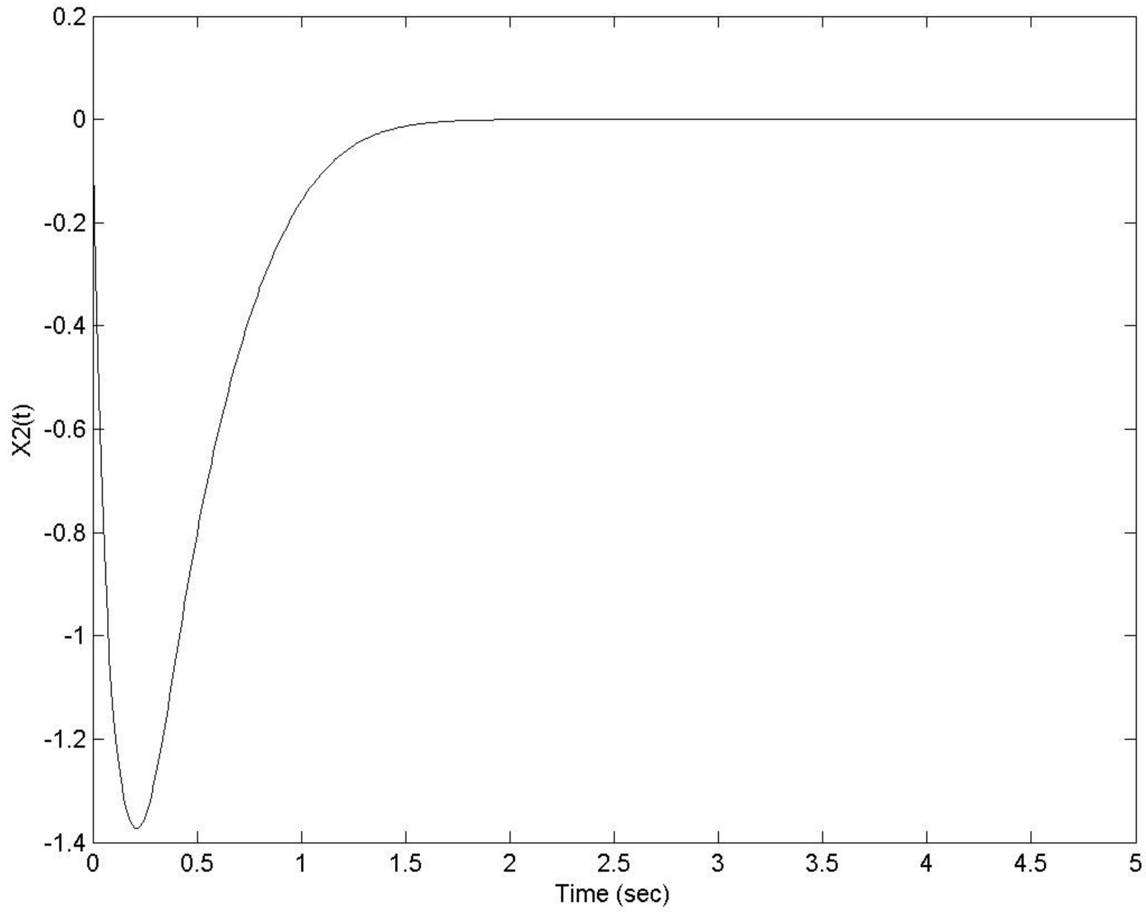


Fig. 3. Response of the angular velocity $x_2(t)$.

Illustrative Example (cont.)

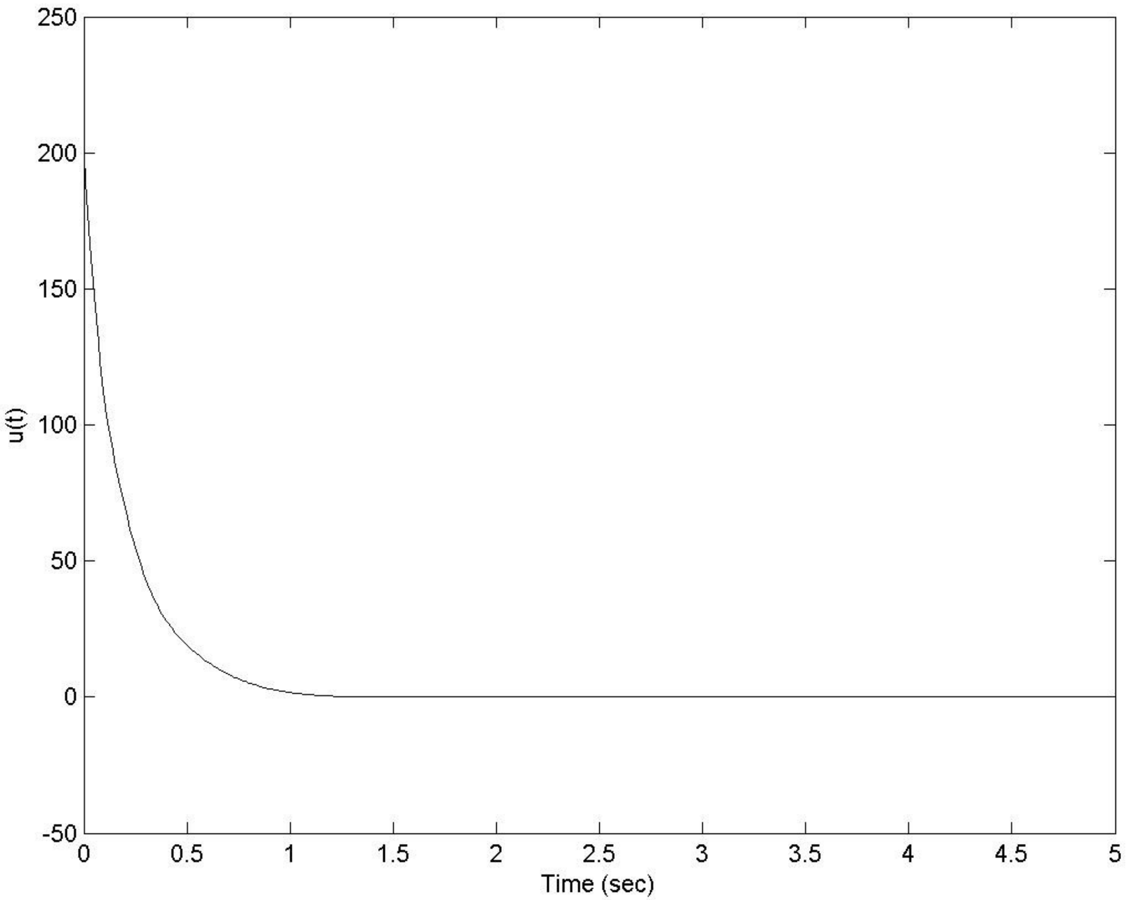


Fig. 4. Control input $u(t)$.

Illustrative Example (cont.)

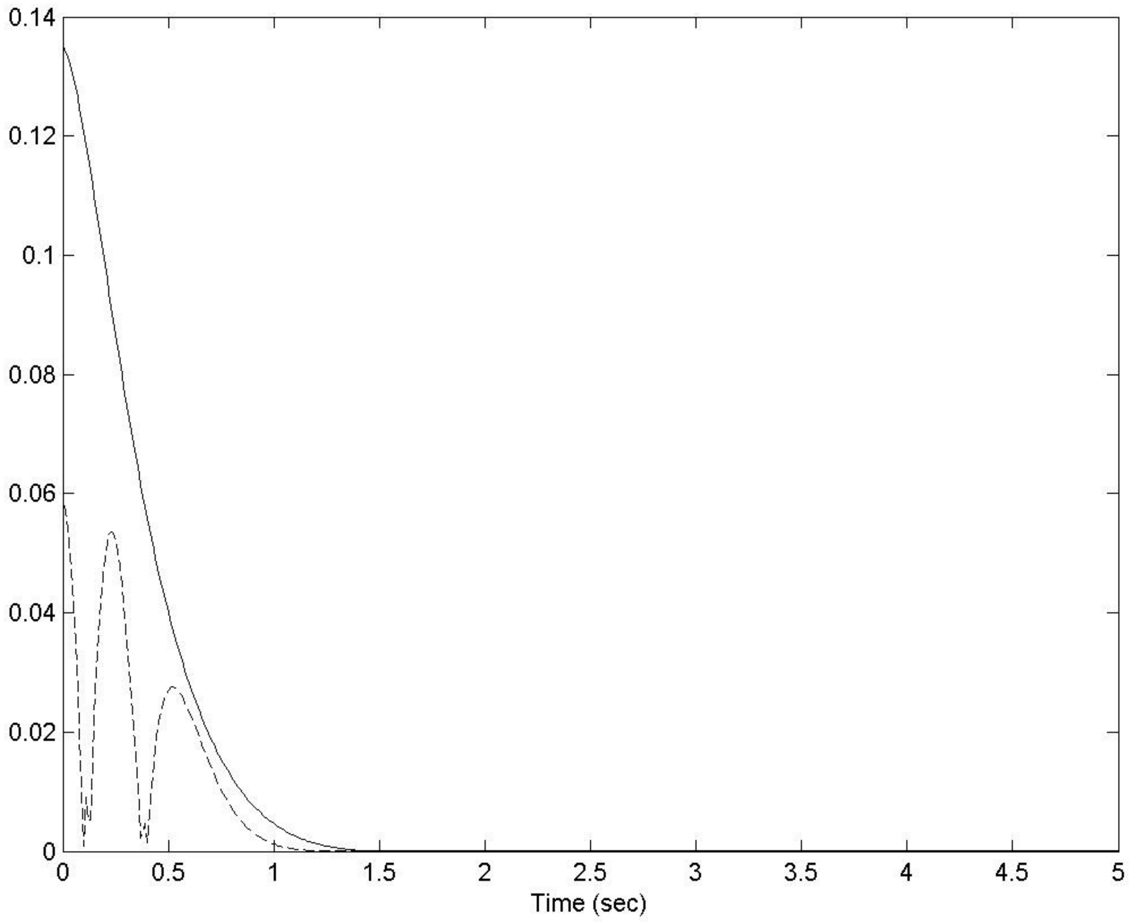


Fig. 5. Plots of $\|\Delta f\|$ (dash line) and $\|A_p x(t)\|$ (solid line).

Illustrative Example (cont.)

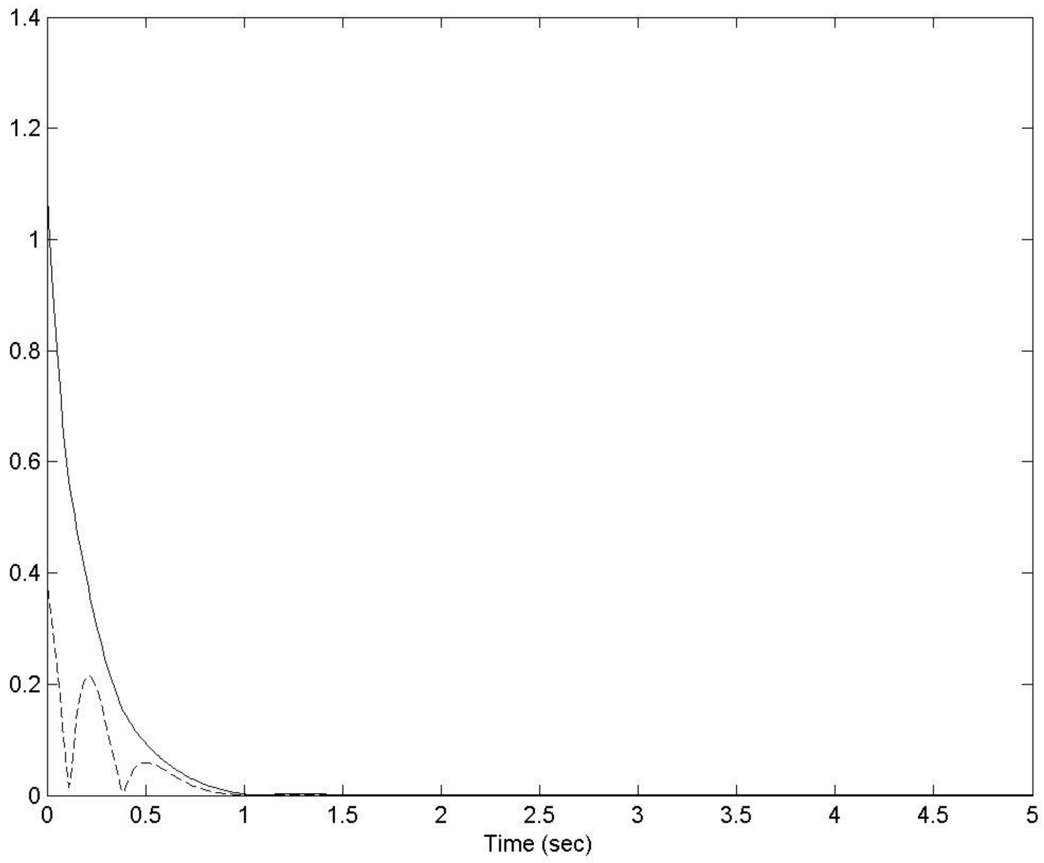


Fig. 6. Plots of $\left\| \Delta g \right\|$ (dash line) and $\left\| \sum_{j=1}^{4} h_j(z(t)) B_p F_j x(t) \right\|$ (solid line).

Conclusions

- The HTGA combines the TGA, which has the merit of powerful global exploration capability, with the Taguchi method, which can exploit the optimum offspring.
- We apply the HTGA approach to solve the function optimization and combinatorial optimization problems and show its capability, feasibility, and robustness.
- The proposed HTGA possesses the merits of global exploration, fast convergence, robustness, and statistical soundness.
- The fusion of soft computing and hard computing can provide the innovative and competitive approaches to solve many problems.

Thank you for your attention.