Multimedia Task Scheduling using Proportion-based Genetic Algorithm

MyungRyun Yoo  Student Member  (Waseda University, mryoo@toki.waseda.jp)
Mitsuo Gen  Member  (Waseda University, gen@waseda.jp)

Keywords : Real-time system, Genetic algorithm, Continuous task, Deadline missing

Video and audio data in multimedia system are referred to as continuous media due to their real-time delivery requirements, whereas text, graphics, and image data are referred to as discrete media. Since continuous media are displayed within a certain time constraint, their computation and manipulation of continuous media should be handled under more limited conditions than that of discrete media. For example, continuous media should be completed before their predetermined deadlines.

Real-time systems are characterized by computational activities with timing constraints and classified into two categories: Hard real-time system and Soft real-time system. In hard real-time system, the violation of timing constraints of certain tasks should not be acceptable. The consequences of not executing a task before its deadline may lead to catastrophic behaviors in certain environments i.e., in patient monitoring system, nuclear plant control, etc. On the other hand, in soft real-time system (ex. telephone switching system, image processing, etc.), in which usefulness of results produced by a task decreases over time after the deadline expires without causing any damage to the controlled environment. The continuous media of multimedia system is typical of the tasks of soft real-time system.

Traditionally, the performance criterion of scheduling algorithms is throughput, utilization of processor, waiting time of tasks, etc. In hard real-time system, the performance of scheduling algorithm is measured by its ability to generate a feasible schedule for a set of real-time tasks. If all the tasks start after their release time and complete before their deadlines, the scheduling algorithm is feasible. Typically, there are Rate Monotonic (RM) and Earliest Deadline First (EDF) derived scheduling algorithms for hard real-time system. They guarantee the optimality in somewhat restricted environments. RM scheduling algorithm assigns static priority to the tasks such that tasks with shorter period get higher priorities. And EDF scheduling algorithm is a dynamic algorithm which schedules the task with the earliest deadline first. There are several RM and EDF derived algorithms for soft real-time system.

Recently, Modified Proportional Share (mPS) Scheduling algorithm is designed for continuous task in soft real-time system by authors. The mPS scheduling algorithm shows better performance than rrPS scheduling algorithm for graceful degradation of performance under the overloaded condition and fewer context switching. However, computational burden and solution accuracy of mPS could be improved by new algorithm based on Genetic Algorithm (GA).

In this paper, we present a new scheduling problem for soft real-time system based on GA. GA has been used already for scheduling problem in manufacturing system such as job shop scheduling, flow shop scheduling and machine scheduling, etc. Especially, GA shows good results in the time constrained task scheduling problem such as real-time task. We focused on scheduling for continuous tasks that are periodic and preemptive. The objective of proposed scheduling algorithm is to minimize the weighted sum of variance of deadline missing and the total number of context switchings among tasks. Some drawbacks (i.e. low resource utilization and avoidable context switching overhead) of RM and EDF derived algorithms for soft real-time systems could be fixed in proposed algorithm. We takes not only advantages of traditional approaches but plus side of GA, such as, high speed, parallel searching and high adaptability.
Multimedia Task Scheduling using Proportion-based Genetic Algorithm

MyungRyun Yoo* Student Member
Mitsuo Gen** Member

Abstract In this paper, we propose a new tasks scheduling algorithm for multimedia task. Multimedia task is typical task of soft real-time system which deadline missing is allowed in. The objective of proposed scheduling algorithm is to minimize the weighted sum of variance of deadline missing and the total number of context switching among tasks. We use proportion-based genetic algorithm for this objective. The effectiveness of the proposed algorithm is shown through a simulation study.

Keywords: Real-time system, Genetic algorithm, Continuous task, Deadline missing

1. Introduction

Recent advances in computing, storage, and communication technologies have made a wide variety of multimedia applications possible. Multimedia systems combine a various information sources, such as audio, video, text, graphics and image, into the wide range of application(1). Video and audio data are referred to as continuous media due to their real-time delivery requirements, whereas text, graphics and image data are referred to as discrete media. Since continuous media are displayed within a certain time constraint, their computation and manipulation of continuous media should be handled under more limited condition than that of discrete media. For example, continuous media should be completed computation before their predetermined deadlines(2).

Real-time systems are characterized by computational activities with timing constraints and classified into two categories: Hard real-time system and Soft real-time system. In hard real-time system, the violation of timing constraints of certain task should not be acceptable(3)(4). The consequences of not executing a task before its deadline may lead to catastrophic consequences in certain environments i.e., in patient monitoring system, nuclear plant control, etc. On the other hand, in soft real-time system (ex. telephone switching system, image processing, etc.), in which usefulness of results produced by a task decreases over time after the deadline expires without causing any damage to the controlled environment(5). The continuous media of multimedia system is typical the task of soft real-time system.

Traditionally, the performance criterions of scheduling algorithms are throughput, utilization of processor, waiting time of tasks, etc. In hard real-time system, the performance of scheduling algorithm is measured by its ability to generate a feasible schedule for a set of real-time tasks. If all the tasks start after their release time and complete before their deadlines, the scheduling algorithm is feasible. Typically, there are Rate Monotonic (RM) and Earliest Deadline First (EDF) derived scheduling algorithms for hard real-time system(6)(7). They guarantee the optimality in somewhat restricted environments. RM scheduling algorithm assigns static priority to the tasks such that tasks with shorter period get higher priorities. And EDF scheduling algorithm is a dynamic algorithm which schedules the task with the earliest deadline first. There are several RM and EDF derived algorithms for soft real-time system. However, these algorithms have some drawbacks to cope with continuous tasks in soft real-time system related resource utilization and pattern of degradation under the overloaded situation. Firstly, in continuous tasks, it is not necessary for every instance of a repetitive task to meet its deadline. For soft real-time system, slight violation of time limits is not so critical. Secondly, RM and EDF scheduling algorithms are required the strict admission control to prevent unpredictable behaviors when the overloaded situation occurs. The strict admission control may cause low utilization of resources(8).

Relatively new researches based on stride scheduler underlying proportional share mechanisms are introduced. They are designed to guarantee the fairness of resource allocation and predictability. Rate Regulating Proportional Share (rrPS) scheduling algorithm based on stride scheduler is proposed for continuous task in soft real-time system. In this research, to specify timing requirements of continuous task, the task’s period and the computation time are reflected. The key concept of rrPS scheduling algorithm is rate regulator which prevents tasks from receiving more resource than its share in a given period. However, rrPS scheduling algorithm does not take account of graceful degradation of performance under the overloaded condition and has the possibility of avoidable context switching overhead(9).

Recently, Modified Proportional Share (mPS) Scheduling algorithm is designed for continuous task in soft real-time system by authors(10). The mPS scheduling algorithm shows better performance than rrPS scheduling algorithm for graceful degradation of performance under the overloaded condition and fewer context switching. However, computational burden and solution accuracy of mPS could be improved by new algorithm based on Genetic Algorithm (GA).

GA has been used already for scheduling problem in manufacturing system such as job shop scheduling, flow shop scheduling and machine scheduling, etc(11)(12). Especially, GA shows good results in the time constrained task scheduling problem such as real-time task(13)(14). In this paper, we focused on scheduling for continuous tasks that are periodic and preemptive. We use proportion-based GA for this task scheduling problem. The objective of proposed scheduling algorithm is to minimize the weighted sum of variance of deadline missing and the total

* Waseda University
  Kitakyushu, 808-0135, Japan
  E-mail: mryoo@toki.waseda.jp
** Waseda University
  Kitakyushu, 808-0135, Japan
  E-mail: gen@waseda.jp
In this study, we assume the followings:
1) All tasks are periodic.
2) All tasks are assigned to same processor (uniprocessor system).
3) All tasks are preemptive.
4) Only processing requirements are significant; memory, I/O and other resource requirements are negligible.
5) All tasks are independent. This means that there are no precedence constraints.
6) The deadline of a task is equal to its period.
7) System is soft real-time system.

A task may be periodic, sporadic, or aperiodic. If task needs to be cyclically executed at constant activation rates, the task is periodic. Sporadic task is not periodic, but may be invoked at irregular intervals and aperiodic task is not periodic and have no upper bound on their invocation rate\(^{(15)}\).

Periodic tasks are characterized by their period and computation time. We focused in continuous media, that is, periodic tasks. Fig. 1 represents characteristics of periodic task. Where, \(i\) is task index, \(i=1,2,\ldots,N\), \(j\) is the index of \(j\)th executed task, \(\tau_i\) is \(i\)th task, \(c_i\) is the computation time of task \(\tau_i\), \(p_i\) is the period of task \(\tau_i\) and \(d_{ij}\) is \(j\)th deadline of task \(\tau_i\).

In hard real-time system, these periodic tasks must be finished their computation before their deadlines. However, since we focused in soft real-time system, rare and slight violence of deadline could be admitted. If a task can not be finished until its next invocation, it is considered as a deadline missing. In actually, in case of playing video frames, 30 frames are played during (PER) 1 second to feel continued motion without halting gap. However, error within quality of service (QoS) of media is not perceived. QoS of video is 10\(^{-1}\) and QoS of audio is 10\(^{-3}\)\(^{(16)}\).

Fig. 2 represents the example of scheduling in soft real-time system, graphically. In Fig. 2, task \(\tau_1\) needs 3 time units as computation time. However, task \(\tau_1\) is not executed really in its first period and is executed during 2 time units in its second period.

**Fig. 1.** Characteristics of continuous tasks

**Fig. 2.** Example of continuous tasks scheduling in soft real-time system
priority(𝜏_𝑖): priority of task 𝜏_𝑖

Equation (1) means that all tasks have their share for executing time of processor based on their period and computation time. The release time of task is defined in Equation (2). Equation (3) means that the deadline of a task is equal to its period as shown Fig. 3. All tasks are cyclically executed at constant activation rates as their periods. In the first stage, the release time of task is 0. From the next stage, the release time of j-th executed task is equal to the deadline of j-th executed task. Equation (4) means that all tasks are preemptive. If the processor is currently occupied by a task and a higher priority task is invoked at the moment, the higher priority task occupy the processor immediately as shown Fig. 3.

3.2 System Constraints Soft real-time system with uniprocessor in this paper can be formulated as follows:

\[ \sum_{i=1}^{N} x_{i} \in \{0, 1\}, \quad \forall \quad t \] .............................. (5)

where

\[ y_{ij} \quad : \quad \text{occurrence of deadline missing at } j\text{th executed time of task } \tau_{i} \]

\[ x_{i} = \begin{cases} 1, & \text{if } \sum_{j=1}^{N} x_{i} < c_{i}, \\ 0, & \text{otherwise} \end{cases} \]

\[ j=1,2,...,n, \quad i=1,2,...,N \] .......................... (6)

where

\[ t \quad : \quad \text{time index}, \quad t=1,2,...,T \]

\[ y_{ij} \quad : \quad \text{occurrence of deadline missing at } j\text{th executed time of task } \tau_{i} \]

\[ x_{i} = \begin{cases} 1, & \text{if } \tau_{i} \text{ is assigned to processor in time } t \\ 0, & \text{otherwise} \end{cases} \]

\[ i=1,2,...,N, \quad t=1,2,...,T \] .......................... (7)

In equation (5), the total number of assigned tasks in time t is 1 or 0. It means that all tasks are assigned to same processor. In other word, the system is uniprocessor system. Equation (6) means system is soft real-time system and slight violence of deadline could be admitted. The value 𝑦_{𝑖}≠0 means that executed time of task during one period is small than its computation time and that is deadline missing.

3.3 Objective Function The problem of scheduling for continuous tasks is to determine execution sequence of tasks with the objective of minimizing the weighted sum of variance of deadline missing and the total number of context switching among tasks.

\[ \min \quad F(\chi) = \alpha \sigma_{m_2}^2 + \beta \mu_{s} \] .......................... (8)

where

\[ \alpha, \beta \quad : \quad \text{coefficients} \]

\[ \alpha + \beta = 1, \quad \alpha, \beta \in [0, 1] \] .......................... (9)

\[ n_{s} \quad : \quad \text{total number of context switching} \]

\[ m_{i} \quad : \quad \text{average deadline missing of } \tau_{i} \]

\[ m_{i} = \frac{1}{n_{i}} \sum_{j=1}^{n_{i}} y_{ij}, \quad i=1,2,...,N \] .......................... (10)

\[ \bar{m} \quad : \quad \text{average of } m_{i} \]

\[ \bar{m} = \frac{1}{N} \sum_{i=1}^{N} m_{i}, \quad i=1,2,...N \] .......................... (11)

\[ \sigma_{m_2}^2 \quad : \quad \text{variance of } m_{i} \]

\[ \sigma_{m_2}^2 = \frac{1}{N} \sum_{i=1}^{N} (m_{i} - \bar{m})^2, \quad i=1,2,...,N \] .......................... (12)

\[ \sigma_{w_2}^2 \quad : \quad \text{worst value of } \sigma_{m_2}^2 \]

\[ \mu_{s} \quad : \quad \text{average of } n_{s} \]

\[ \mu_{s} = \frac{n_{s}}{T} \] .......................... (13)

\[ \mu_{w} \quad : \quad \text{worst value of } \mu_{s} \]

4. Substantial Components of GA

In this paper, several new techniques are proposed in the encoding and decoding algorithm of genetic string and the genetic operations are introduced for discussion. They are explained in the following subsections.

4.1 Encoding and Decoding A chromosome 𝑉_\chi = \{𝑣_\chi \}, \quad k=1,2,...,\text{popSize}, \quad \text{represents tasks which are assigned to a processor and } r\text{th gene } v_\chi \text{ represents the task which is assigned to a processor in time } t. \quad \text{Where } \text{popSize} \text{ is total number of chromosomes in each generation. The length of a chromosome is } T. \quad \text{It means the minimum term of scheduling. Fig. 4 represents the structure of chromosome for proposed genetic algorithm during } 8 \text{ time units.}

Encoding and Decoding procedures can be explained as:

procedure 1: Proportion-based encoding

step 1: Calculate 𝑦_\chi \quad \text{and } \gamma_\chi, \quad \gamma_\chi \quad \text{is total sum of } \gamma_\chi.

step 2: Calculate 𝑞_\chi. 𝑞_\chi \quad \text{is the cumulative rate of share for executing time of processor and equation (14).}

\[ q_\chi = \begin{cases} \frac{y_\chi}{\gamma_\chi}, & i = 1 \\ \frac{y_\chi - q_{i-1}}{\gamma_\chi}, & i = 2,3,...,N \end{cases} \] .......................... (14)

step 3: Generate a random number 𝑟 \text{ from range } [0,1]. \quad \text{if } 𝑟 ≤ 𝑞_\chi, \text{ then assign } 1 \text{ to } v_\chi, \text{ else if } 𝑞_{i-1} < 𝑟 ≤ 𝑞_\chi, \quad i=2,3,...,N, \text{ then assign } i \text{ to } v_\chi.

\[ \text{schedule} \] .......................... (15)

\[ \text{Chromosome } v_\chi \]

\[ v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \quad v_7 \quad v_8 \]

\[ T \] .......................... (16)

Fig. 3. Characteristics of continuous tasks II

Fig. 4. Chromosome representation
4.2 Evaluation Function and Selection

The selection is the main method for GA mimics evolution in natural systems: fitter an individual is, the highest is its probability to be selected. The commonly used method is roulette wheel selection (12). Parents are selected in the current population. Parents chromosomes are created by mating, with probability \( p_c \) and selecting.

The evaluation function is essentially the objective function in the problem. It provides a means of evaluating the search node and it also controls the selection process (12). The fitness evaluation method options considered were “Fitness is Evaluation” (FE), Windowing (WD) and Linear Normalization (LN) (19, 20). In this paper, we compare with the results of these evaluation methods.

FD is the most basic fitness evaluation method. Because we use the roulette wheel selection, we convert the minimization problem to maximization problem (21), that is, the used evaluation function with FE eval\(_{FD} \) is then

\[
eval_{FD}(V_i) = 1 / F(x), \quad \forall \ k \quad \text{if} \quad \eval_{FD}(V_i) \leq \eval_{FD}(V_i')
\]

WD is a technique for assigning fitness to a population of chromosomes to boost the fitness of the weaker member, in order to prevent their elimination and the resulting dominance by a small number of chromosomes. It works in the following manner: Find the minimum evaluation in the population. Assign each chromosome a fitness equal to the amount that it exceeds this minimum. Optionally, a minimum amount greater than the minimum value may be created as a guard against the lowest chromosomes having no chance of reproduction. Parents are selected by roulette wheel selection.

In this paper, we define evaluation function with WD eval\(_{WD} \) as follows:

\[
eval_{WD}(V_i) = \eval_{FD}(V_i) - \min(\eval_{WD}(V_i)) \quad \forall \ k \quad \text{if} \quad \eval_{WD}(V_i) > \eval_{WD}(V_i')
\]

LN is fitness evaluation method intended to increase competition between similar individuals. It works in the following manner: Each individual’s fitness is calculated with the fitness function. The individuals are ordered from least to most fit. Each individual’s fitness becomes its order number. Parents are selected by roulette wheel selection. In this paper, chromosomes are ordered in increasing order of their \( F(x) \). The evaluation function with LN eval\(_{LN} \) for each individual is defined as follows:

\[
eval_{LN}(V_i) = \begin{cases} \text{popSize}, & k = 1 \\ \eval_{WD}(V_i') - 1, & k = 2, 3, ..., \text{popSize} \\ \end{cases}
\]

where \( k' \) is the ordered number of chromosome \( V_i \).

4.3 Genetic Operators

We propose Period Unit Crossover (PUX). This operator creates two new chromosomes (the offspring) by mating two chromosomes (the parents), which are combined as shown Fig. 5: the period (from \( r_0 \) to \( d_0 \)) is selected by \( i, j \) randomly chosen, and each offspring chromosomes are built by exchanging the substrings of selected period between parents. \( V'_1 \) and \( V'_2 \) mean offspring 1 and offspring 2 each other.

The procedure of PUX will be follows as:

**procedure 2: Proportion-based decoding**

**step 1:** Derive \( x_i \) from \( V_i \).

**step 2:** Calculate \( n_i^j \). \( n_i^j \) is executed time of task \( t_i \) from \( r_j \) to \( d_j \).

**step 3:** If \( n_i^j > c_i \), then regulate executed time by changing the value 1 of randomly selected locus to 0.

**step 4:** If \( n_i^j < c_i \), then assign \( i \) to processor in idle time.

**step 5:** Generate schedule by appending task number \( i \) with value 1 at time \( t_i \).

**4.4 Reproduction and Population Replacement**

During reproduction and replacement steps, offspring chromosomes are created by mating, with probability \( p_c \), pairs of parents selected in the current population. Parents chromosomes are replaced by this offspring. Then chromosomes are mutated with probability \( p_m \), randomly using one of the mutation operator (12). New population is built through evaluating chromosomes and selecting.

This iterative evolution process is stopped as soon as one solution is found. However, we limit the number of offspring produced to \( \text{maxGen} \), in order to avoid prohibitive calculation time, and to ensure that the GA will stop when treating an infeasible problem. Consequently, our proportion-based GA obeys to the following algorithm:

**algorithm : Proportion-based GA (PGA)**

**step 1:** Generate the popSize chromosomes of the initial population using proportion-based encoding (procedure 1) and evaluate them using proportion-based decoding (procedure 2).

<table>
<thead>
<tr>
<th>( i )</th>
<th>( c_i )</th>
<th>( p_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>parents</th>
<th>offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_1 )</td>
<td>( V_1' )</td>
</tr>
<tr>
<td>( V_2 )</td>
<td>( V_2' )</td>
</tr>
</tbody>
</table>

Fig. 5. Period unit crossover (PUX)

<table>
<thead>
<tr>
<th>parent</th>
<th>offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V' )</td>
<td>( V'' )</td>
</tr>
</tbody>
</table>

Fig. 6. Altering mutation
step 2: Produce offspring chromosomes using PUX (procedure 3), with probability $p_C$, pairs of parents chromosomes selected in the population.

step 3: Mutate chromosomes with the probability $p_M$.

step 4: Evaluate chromosomes using the proportion-based decoding (procedure 2).

step 5: Select chromosomes based on roulette wheel selection.

step 6: Repeat from steps 2-5 to this step until finding the chromosome with 0 value of fitness or maxGen.

5. Numerical Results

For the validation of Proportion-based GA (PGA), several numerical tests are performed. We compared PGA with EDF and mPS scheduling algorithms.

EDF scheduling algorithm is a dynamic algorithm which schedules the task with the earliest deadline first and guarantee the optimality in hard real-time system.

mPS Scheduling algorithm is designed for soft real-time system by the first author.

Numerical tests are performed with three task sets. Table 1, Table 2 and Table 3 represent small scale, medium scale and large scale tasks sets respectively. For tasks’ computation time and deadline, we use random number based on exponential distribution. They are different in total number of tasks and the degree of overloaded state.

Table 4 represents the results of numerical tests with the task set 1, 2 and 3. The parameters were set to 0.7 for crossover ($p_C$), 0.3 for mutation ($p_M$), and 30 for population size ($popSize$). Probabilities for crossover are tested from 0.5 to 0.8, from 0.001 to 0.4 for mutation, with the increments 0.05 and 0.001 respectively. For population size, individuals from 20 to 200 are tested. Table 4 shows the best computation result selected from number of computation results with various combinations of parameters.

Table 4 shows that the value of objective function $F(x)$ value in proposed algorithm (PGA) is smaller than those of EDF and mPS algorithms. The objective of proposed scheduling algorithm is to minimize the weighted sum of variance of deadline missing and the total number of context switching among tasks. Therefore, we can see that the performance of proposed PGA is better than those of other 2 algorithms in our test cases. Also, $\alpha$ and $\beta$ can be adjust for the important point. In this paper, $\alpha$ and $\beta$ are setted equally. In PGA, results are different accordingly evaluation method. In the result of simulation, evaluation method with LN is better than FE and WD for stability. The evaluation method with LN fitted to evaluate individuals with similar fitness value. In this continuous tasks scheduling problem, the fitness values of individuals is scarcely different. It is caused that the variance of deadline missing is very small value.

6. Conclusions

A new tasks scheduling algorithm PGA is proposed in this paper. The proposed algorithm uses GA and is designed for continuous media in soft real-time system. In soft real-time system, deadline missing can be allowed. PGA is to determine execution sequence of tasks with the objective of minimizing the weighted sum of variance of deadline missing and the total number of context switching among tasks. From the numerical results, the results of the PGA is better than EDF and mPS scheduling algorithms. Also, three evaluation methods are compared in PGA. The evaluation method with LN is better than other evaluation methods. The evaluation method with LN fitted to evaluate individuals with similar fitness value. In this continuous tasks scheduling problem, the fitness values of individuals is scarcely different. It is caused that the variance of deadline missing is very small value.

This determines the next step of our study. We plan to design tasks scheduling algorithm in multiprocessor soft real-time system using proportion-based GA and to design tasks scheduling algorithm in hard real-time system.

Acknowledgments

This work is partly supported by Waseda University Grant for Special Research Projects 2004 and the Ministry of Education, Science and Culture, the Japanese Government: Grant-in-Aid for Scientific Research (No.17510138).

(Manuscript received April, 4, 2004, revised Nov. 11, 2005)

References


(2) P. J. Tsun and H. Babak : “Dynamic real time scheduling strategies for


MyungRyun Yoo (Student Member) received B.E from Andong National University, Korea in 1994, M.S from Pohang University of Science & Technology, Korea in 1996 and Ph.D. degree from YeoungNam University, Korea in 2002. She joined Andong Information Technical Junior College as Assit. Professor in 1996. She is currently a doctor course student of Graduate School of Information, Production & Systems, Waseda University. Her research field is Genetic Algorithms, Real-Time System, Scheduling, Multimedia System, etc.

Mitsuo Gen (Member) received B. S., M. S. and Ph.D. degree from Kogakuin University, Japan in 1969, 1971 and 1974, respectively. He was Lecturer (1974-1981) ; Assoc Professor (1981-1988); Professor (1988.4-2003.3) at Ashikaga Institute of Technology, Japan. He is currently a Professor of Graduate School of Information, Production & Systems, Waseda University. He was Visiting Assoc. Prof. at University of Nebraska-Lincoln, USA in 1981-1982; Visiting Prof. at University of California at Berkeley, USA in 1999-2000. His research interests include Genetic Algorithms, Neural Network, Fuzzy Logic, and the applications of evolutionary techniques to Network Design, Scheduling, Reliability Design. He has authored *Genetic Algorithms & Engineering Design*, Wiley, NY (1997), *Genetic Algorithms & Engineering Optimization*, Wiley, (2000) with Dr. Runwei Cheng. and is an Area Editor: *Computers & Industrial Engineering*. 