

# 수요를 고려한 잔고의 최적 Capacity 확장을 위한 진화 프로그램

조 정 복<sup>†</sup> · 양 황 규<sup>†</sup> · 이 재 옥<sup>†</sup> · 함 창 현<sup>††</sup> · Boading Liu<sup>†††</sup>

## 요 약

본 논문에서는, 수요와 가격의 다양한 변화하에서 언제 그리고 얼마나 많은 용적이 추가되어야 하는지를 결정하는, 수요를 고려한 잔고의 최적 용적 확장 문제를 위한 진화프로그램을 제시한다. 국부적 수렴 문제를 극복하기 위하여, 우리는 지수-적합 계수 방법을 적용한다. 염색체 표현의 개선을 위하여, 새로운 영역을 산출하고 가능한 해를 생성하는 이형-영역 돌연변이와 계단형의 음의 방향으로 염색체를 돌연변이 시키는 동형-영역 돌연변이를 도입한다. 마지막으로, 수치적 실험 결과를 제시한다.

## Evolution Program for Optimal Capacity Expansion with Backlogs in Demand

Jung-Bok Jo<sup>†</sup> · Hwang-Kyu Yang<sup>†</sup> · Jae-Uk Lee<sup>†</sup> · Chang-Hyun Ham<sup>††</sup> · Boading Liu<sup>†††</sup>

## ABSTRACT

This paper presents an evolution program for optimal capacity expansion problem with backlogs in demand which is to determine when and how much capacity should be added under varying types of demand and cost. To overcome premature convergence and stalling, we employ an exponential-fitness scaling scheme. To improve the chromosomes, we introduce hetero-dimensional mutation which generates a new dimension and produces a feasible solution, and homo-dimensional mutation which mutates the chromosomes in the negative of gradient direction. Finally, a numerical example is discussed.

### 1. Introduction

Capacity expansion problems have been studied over thirty years and applied in a wide variety of areas, for example, heavy industries, communication networks, electrical power services and water resource systems. The purpose of capacity models is

to determine when and how much capacity should be added under varying types of demand and cost. Usually, there exist two types of demand, deterministic and stochastic. The optimal capacity expansion with deterministic demand is reported by [14], [2], [3], [6] and summarized by [13]. Stochastic case is considered by [5], [1] and [4], etc.. In this paper we will restrict our attention to deterministic case with continuous time. On the other hand, evolution programs are a type of stochastic search methods and have been used widely for numerical

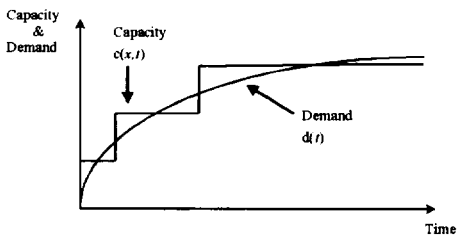
† 정 회 원 : 동서대학교 정보시스템공학부 교수  
†† 정 회 원 : 경남정보대학 컴퓨터정보학부 교수  
††† 미 회 원 : Department of Applied Mathematics, School of Science, Tsinghua University, Professor  
논문접수: 1998년 7월 28일, 심사완료: 1998년 11월 24일

optimization problem([11], [15]), such as optimal control problems, transportation problems, traveling salesman problems, drawing graphs, scheduling and machine learning. Evolution programs are also studied by several researchers and summarized by [7]. As a research work related closely with this paper, Gen and Liu[9] presented an evolution program for continuous time production plan problem which is to determine a rate of production under varying types of demand and cost. An evolution program for optimal capacity expansion problem has been reported by [10] when shortage of supply is not allowed.

In this paper we present an evolution program for optimal capacity expansion with backlogs in demand which is to determine when and how much capacity should be added under varying types of demand and cost. To overcome premature and stalling, we suggest an exponential-fitness scaling scheme. To improve the chromosomes, we introduce nonhomo-dimensional mutation which generates a new dimension and produces a feasible solution, and homo-dimensional mutation which mutates the chromosomes in the negative of gradient direction. Finally, some numerical examples are discussed.

## 2. Capacity Expansion with Backlogs

The criterion used here for determining the best capacity policy is to minimize the discounted costs associated with the expansion process, including costs for expansions, shortages, congestion, idle capacity, maintenance and inventory, taking into physical constraints. In this paper, we consider the following continuous time case with backlogs in demands.



(Fig. 1) Capacity Expansion with Backlogs in Demand

(Fig. 1) charts the curve of demand  $d(t)$ , i.e., a function that specifies the demand at time  $t$  and expansion of capacity over time under the following simplifying assumptions :

- (i) the equipment has an infinite economic life;
- (ii) whenever the demand catches up with the existing capacity,  $x$  units of new capacity are installed;
- (iii) the planning horizon is transacted after an arbitrary finite time.

The installation cost that results from a single capacity increment of size  $x$  at time  $t$  is assumed to be given by a function  $c(x, t)$  which is increasing in  $x$  and decreasing in  $t$ . The shortage cost that results from a shortage quantity  $y$  at time  $t$  is assumed to be given by a function  $s(y, t)$  which is also increasing in  $y$  and decreasing in  $t$ .

Our problem is to determine an integer  $n$  and  $0 = t_1 < t_2 < \dots < t_n < T$  such that the total installation cost is minimizes, i.e.,

$$\min_n \min_{0=t_1 < t_2 < \dots < t_n < T} \sum_{i=1}^n c(d(t_{i+1}) - d(t_i), t_i) \quad (1)$$

where  $t_i$  denotes the time when the capacity is expended  $i$  times according to a change in demand and  $T$  denotes the upper limited time to consider a capacity expansion.

However, the previous studies of optimal expansion of capacity only discuss the discrete time case. The computing method for continuous cases was not reported by any author. Maybe it is very difficult to design a traditional approach to solve such a problem. This is the basic motive to develop an evolution program.

From the basic eq.(1), we see that this problem is very different from the classical optimization because that both of the dimension  $n$  and decision vector  $t = (t_1, t_2, \dots, t_n)$  are variables, but the classical cases have fixed dimension.

### 3. Evolution Program

In this section we present an evolution program to optimal capacity expansion problem. We will discuss representation structure, initialization process, evaluation function, selection, mutation and crossover operations.

#### 3.1 Representation Structure

We use an integer  $n$  and a vector  $V=(x_1, x_2, \dots, x_{2n})$ ,  $\epsilon \leq x_i \leq 1$  as a chromosome to represent a solution  $((t_1, y_1), (t_2, y_2), \dots, (t_n, y_n))$  to the optimal capacity expansion problem, where  $\epsilon$  is an appreciate small positive number. The mapping from a chromosome  $(x_1, x_2, \dots, x_{2n})$  into a solution  $((t_1, y_1), (t_2, y_2), \dots, (t_n, y_n))$  can be written as follows :

$$\begin{aligned}
 t_1 &= 0 \\
 t_i &= \frac{x_2 + x_4 + \dots + x_{2i-2}}{x_2 + x_4 + \dots + x_{2n}} \cdot T, \quad i = 2, 3, \dots, n \quad (2) \\
 y_i &= x_{2i}, \quad i = 1, 2, \dots, n
 \end{aligned}$$

It follows from eq.(2) that the solution  $(t_1, t_2, \dots, t_n)$  is feasible.

#### 3.2 Initialization Process

We define an integer  $pop\_size$  as the number of chromosomes.  $pop\_size$  chromosomes will be randomly initialized by the following way.

To determine a chromosome, we firstly determine its dimension, i.e., the number of capacity expansions. Usually, the range for possible  $n$  can be given by some prior knowledge. So we can generate a random integer  $n$  in that range, then  $2n$  random numbers  $x_1, x_2, \dots, x_{2n}$  on the interval  $[\epsilon, 1]$ . A chromosome is completed, i.e.,  $(x_1, x_2, \dots, x_{2n})$ . Repeat this process  $pop\_size$  times and produce  $pop\_size$  initial feasible solutions.

#### 3.3 Evaluation Function

Let  $u$  denote the original fitness, i.e., the objec-

tive value. Usually, the original fitness proportionate-reproduction scheme frequently causes two significant difficulties : premature convergence termination at early generations, and stalling at late generations. To overcome these two problems, Goldberg[11] suggests a linear-fitness scaling scheme,  $u' = au + b$ , Lee and Johnson[12] suggests a modified linear-fitness scaling scheme, i.e., for the lower-fitness strings, the minimum original fitness  $u_{min}$  maps to 0; for the higher-fitness strings, the maximum original fitness  $u_{max}$  maps to a parameter  $f_{multiple}$  which is greater than 1; the average fitness  $u_{avg}$  maps to 1. Michalewicz[15] suggests a power law scaling scheme,  $u' = u^k$ , where  $k$  is a real number which is close to one.

Here we will employ an exponential-fitness scaling scheme suggested by Gen et al.[8] because this scaling scheme is a combination form of the left and right extinctive selections to prevent both poor and super chromosomes from reproduction[17]. We define three preference parameters  $p_1, p_0$  and  $p_2$  ( $0 < p_1 < p_0 < p_2 < 1$ ) which can determine three critical numbers  $u_1, u_0$  and  $u_2$  such that there are  $(p_1 \cdot pop\_size), (p_0 \cdot pop\_size)$  and  $(p_2 \cdot pop\_size)$  chromosomes which are less than  $u_1, u_0$  and  $u_2$ , respectively, in the set of the  $pop\_size$  chromosomes at the current iteration.

For the optimal capacity expansion, the original fitness  $u_1$  maps to  $2 - e^{-1} \approx 1.63$ , the original fitness  $u_0$  maps to 1, and the original fitness  $u_2$  maps to  $e^{-1} \approx 0.37$ . Then the relation between the original fitness  $u$  and the exponential fitness  $u'$  is

$$u' = \begin{cases} 2 - \exp\left[-\frac{u - u_0}{u_1 - u_0}\right], & u < u_0 \\ \exp\left[-\frac{u - u_0}{u_2 - u_0}\right], & u \geq u_0 \end{cases} \quad (3)$$

which is shown in (Fig. 2). Obviously, the preference parameters  $p_1$  and  $p_2$  are used to designate the chromosomes with extreme values to

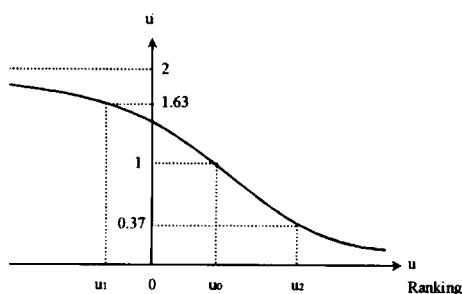
be eliminated.

Thus we can define the evaluation function as follows :

$$eval(V) = u' / \sum_{i=1}^{pop\_size} u_i' \quad (4)$$

where  $V$  is a chromosome,  $u'$  is the exponential fitness of  $V$ , and  $u_i'$  is the exponential fitness of chromosome  $V_i$ ,  $i = 1, 2, \dots, pop\_size$ , respectively.

For the above mentioned evaluation function, the chromosome with higher fitness can have more chance to produce offsprings.



(Fig. 2) Exponential-fitness Function

### 3.4 Selection Process

The selection process is based on spinning the roulette wheel  $pop\_size$  times, each time we select a single chromosome for a new population in the following way :

- (a) Calculate the cumulative probability  $q_i$  for each chromosome  $V_i$ ,

$$q_0 = 0$$

$$q_i = \sum_{j=1}^i eval(V_j), \quad i = 1, 2, \dots, pop\_size. \quad (5)$$

- (b) Generate a random real number  $r$  in  $[0, 1]$ .
- (c) Select the  $i$ -th chromosome  $V_i$  ( $1 \leq i \leq pop\_size$ ) if  $q_{i-1} < r \leq q_i$ .
- (d) Repeat steps (b) and (c)  $pop\_size$  times and obtain  $pop\_size$  copies of chromosomes.

### 3.5 Crossover Operation

We define a parameter  $P_c$  of an evolution sys-

tem as the probability of crossover. This probability gives us the expected number  $P_c \cdot pop\_size$  of chromosomes which undergo the crossover operation.

Firstly we generate a random real number  $\gamma$  in  $[0, 1]$ ; secondly, we select the given chromosome for crossover if  $\gamma < P_c$ . Repeat this operation  $pop\_size$  times and produce  $P_c \cdot pop\_size$  parents, averagely.

For each pair of parents (vectors  $V_1$  and  $V_2$ ), if they have the same dimension, the crossover operator on  $V_1$  and  $V_2$  will produce two children  $X$  and  $Y$  as follows :

$$X = c_1 \cdot V_1 + c_2 \cdot V_2 \quad (6)$$

$$Y = c_2 \cdot V_1 + c_1 \cdot V_2$$

where  $c_1, c_2 \geq 0$  and  $c_1 + c_2 = 1$ . Otherwise, skip the crossover operation.

Since the constraint set is convex, this arithmetical crossover operation ensures that both children are feasible if both parents are.

### 3.6 Mutation Operation

Two types of mutation operators are defined in this evolution program. The first, *hetero-dimensional mutation*, generates a new dimension and then produces a feasible solution. The second, *homo-dimensional mutation*, mutates the chromosomes in the negative of gradient direction.

It is well-known that the Taylor's expansion of a continuous differentiable function  $f$  is

$$f(x + \Delta x) = f(x) + (\nabla f(x + \theta \Delta x))^T \Delta x, \quad x \in R^n \quad (7)$$

where  $0 \leq \theta \leq 1$ ,  $\nabla f(x)$  denotes the gradient of the function  $f$  at the point  $x$ ,  $\Delta x$  is a small perturbation in  $R^n$ . So for the optimal capacity expansion, it may be better to choose the negative direction of gradient as the mutation direction.

Because of the complexity of the problem, we will calculate the  $i$ -th component of the gradient

(regardless of the existence) approximately by

$$\frac{f(x_1, \dots, x_i + \Delta x_i, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{\Delta x_i} \quad (8)$$

where  $\Delta x_i$  is a small real number.

We define two parameters  $P_{m1}$  and  $P_{m2}$  of an evolution system as the probabilities of hetero-dimensional mutation and homo-dimensional mutation. Then the two probabilities give us the expected numbers  $P_{m1} \cdot pop\_size$  and  $P_{m2} \cdot pop\_size$  of chromosomes which undergo the hetero-dimensional mutation and homo-dimensional mutation, respectively.

Generating a random real number  $\gamma$  in  $[0, 1]$ , we select the given chromosome for hetero-dimensional mutation and homo-dimensional mutation if  $\gamma < P_{m1}$  and  $P_{m1} < \gamma < P_{m1} + P_{m2}$ , respectively. Let a parent denoted by a vector  $V = (x_1, x_2, \dots, x_n)$ , be selected. For hetero-dimensional mutation, we generate a new dimension  $n$  and then produce an  $n$ -dimensional feasible solution by the procedure initialization. For homo-dimensional mutation, we choose a direction  $d$  which is a negative direction of the approximate gradient and a large number  $M$  which ensures the operator is probabilistically complete, if  $V + M \cdot d$  is not feasible, then we set  $M$  as a random number between 0 and  $M$  until it is feasible, thus the offspring is

$$\hat{V} = V + M \cdot d \quad (9)$$

Repeat this operation  $pop\_size$  times and produce  $(P_{m1} + P_{m2}) \cdot pop\_size$  offsprings, averagely.

### 3.7 Algorithm

Following selection, crossover and mutation, the new population is ready for its next evaluation. The evolution program will terminate after a given number of cyclic repetitions of the above steps. We can summarize the evolution program for optimal capacity expansion as follows.

Step 0. Set parameters

```
input maxgen; // numbers of generations
pop_size; // population size
P_m1; // probability of hetero-
           dimensional mutation
P_m2; // probability of homo-
           dimensional mutation
P_c; // probability of crossover
p_1, p_0, p_2; // preference parameters
```

Step 1. Initialization process

```
for i = 1 to pop_size do
    produce a random integer n;
    produce n random real numbers
        x_1, x_2, ..., x_n on [ε, 1];
    V_i = (x_1, x_2, ..., x_n);
endfor
```

Step 2. Evaluation

```
for i = 1 to pop_size do
    compute the objectives u_i for V_i;
endfor
for i = 1 to pop_size do
    compute the exponential-fitness u_i';
endfor
for i = 1 to pop_size do
    compute the cumulative probabilities
        q_i = \sum_{j=1}^i u_j' / \sum_{j=1}^{pop\_size} u_j';
endfor
```

Step 3. Selection operation

```
for i = 1 to pop_size do
    if ( q_{i-1} < random() <= q_i ) then
        select V_i;
    endif
endfor
```

Step 4. Crossover operation

```
for i = 1 to \frac{pop\_size}{2} do
```

```

if ( random( ) <= Pc ) then
    int j = random( pop_size);
    int k = random( pop_size);
    perform the crossover on j-th and
    k-th chromosomes;
endif
endfor

```

Step 5. Mutation operation

```

for i = 1 to pop_size do
    r = random( );
    if ( γ <= Pm1 ) then
        hetero_dimensional_mutation( );
    else
        if ( Pm1 < γ <= Pm1 + Pm2 ) then
            homo_dimensional_mutation( );
        endif
    endif
endif
endfor

```

Step 6. Termination test

```

if ( number of iterations < maxgen) then
    goto step 2;
else
    stop;
endif

```

4. Numerical Experiments

The program for the evolution program has been written in C language. We will use it to solve the following numerical example which is a modification of one in [16].

We take the demand pattern as

$$d(t) = b \cdot \sin\left(\frac{t}{2T} \pi\right), \quad 0 \leq t \leq T \quad (10)$$

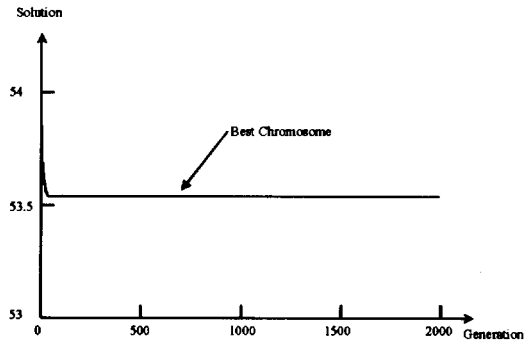
which will be satisfied by the production capacity and the cost function as

$$c(q, t) = \begin{cases} 0, & q = 0 \\ \exp[-at] \cdot (k + e \cdot q), & q > 0 \end{cases} \quad (11)$$

which results from a single capacity expansion of size  $q$  at time  $t$ .

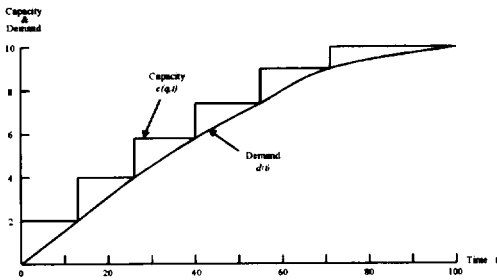
In this numerical example, we set the parameters as  $b = 10$ ,  $T = 100$ ,  $a = 0.06$ ,  $k = 10$ ,  $e = 10$ .

In our experiment, the population size is 50, the probability of crossover is 0.1, the probability of hetero-dimensional mutation is 0.1, the probability of homo-dimensional mutation is 0.4, the preference parameters  $p_1$ ,  $p_0$  and  $p_2$  are defined as 0.1, 0.5 and 0.9, respectively.



(Fig. 3) Progress of Evolution

(Fig. 3) shows that in 10 iterations the best solution of problem is 53.594822; in 100 iterations the best solution is 53.540710; in 500 iterations the best solution is 53.510918; in 1000 iterations the best solution is 53.510906; in 2000 iterations the best solution is 53.510899. The process of capacity expansion is shown by (Fig. 4) in which the step-like function represents the production capacity  $c(q,t)$  and the curve represents the demand  $d(t)$ . The curve of demand  $d(t)$  can be obtained by eq.(10) when we set the parameters as  $b=10$  and  $T=100$ . Also, (Fig. 4) shows that the policy is to add 1.995217 at the beginning, 1.959452 at time 12.454445, 1.870075 at time 25.883461, 1.714847 at time 39.583107, 1.459610 at time 54.371712, and 1.000799 at time 71.275131 with six times capacity expansions. In performing the algorithm with 1000 generations, the CPU time spent for the evolution program is 76.3 seconds on a NEC EWS4800/210II workstation.



(Fig. 4) Process of Capacity Expansion

The proposed evolution program has been performed on a lot of optimal capacity expansion problems for different parameters. All of the results are satisfactory for the same evolution parameters. For example, when we set the parameters as  $b = 15$ ,  $T = 100$ ,  $a = 0.05$ ,  $k = 20$ ,  $e = 15$ , a run of our evolution program with 2000 iterations shows that the optimal cost is 128.452 with six times expansions at

$$(t_1, t_2, t_3, t_4, t_5, t_6) = (0.000, 13.398, 27.103, 41.447, 56.850, 74.252).$$

If we set  $b = 8$ ,  $T = 100$ ,  $a = 0.05$ ,  $k = 20$ ,  $e = 8$ , a run of our evolution program with 2000 iterations shows that the optimal cost is 59.293 with three times expansions at

$$(t_1, t_2, t_3) = (0.000, 23.518, 49.835).$$

### 5. Conclusion

In this paper we present an evolution program for continuous time optimal capacity expansion problem. Usually, the fitness proportionate-reproduction scheme frequently cause two significant difficulties: premature convergence at early generations, and stalling at the late generations. We employed an exponential-fitness scaling scheme to overcome the two problems as above.

To improve the chromosomes, we introduce hetero-dimensional mutation which generates a new dimension and produces a feasible solution, and

homo-dimensional mutation which mutates the chromosomes in the negative of gradient direction.

Finally, we discuss a numerical example which shows that the evolution program is an effective algorithm. For optimization problems, the time necessary for an algorithm to converge to the optimum depends on the number of decision variables, the time complexity of most algorithms can increase geometrically with the number of variables, however, the time complexity of the proposed evolution program increases linearly as the dimension increases.

### References

- [1] J. A. Buzacot, A. B. Chaouch, "Capacity expansion with interrupted demand growth," *EJOR*, Vol.34, pp.19-26, 1988.
- [2] D. Erlenkotter, "Sequencing expansion projects," *Operations Research*, Vol.21, pp.542-553, 1973.
- [3] D. Erlenkotter, "Capacity expansion with imports and inventories," *Management Science*, Vol.23, pp.694-702, 1977.
- [4] D. Erlenkotter, S. Sethi and N. Okada, "Planning for surprise : water resources development under demand and supply uncertainty I, the general model," *Management Science*, Vol.35, pp.149-163, 1989.
- [5] J. Freidenfelds, "Capacity expansion when demand is a birth-death process," *Operations Research*, Vol.28, pp.712-721, 1980.
- [6] J. Freidenfelds, "Capacity Expansion, Analysis of Simple Models with Applications," Elsevier, New York, 1981.
- [7] D. B. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Transactions on Networks*, Vol.5, pp.3-14, 1994.
- [8] M. Gen, B. Liu and K. Ida, "Evolution program for deterministic and stochastic optimization," *European Journal of Operational Research*, Vol. 94, pp.618-625, 1996.
- [9] M. Gen, B. Liu, "Evolution program for production plan problem," *Engineering Design and*

Automation, Vol.1, pp.199-204, 1995.

- [10] M. Gen, B. Liu, "Evolution program for optimal capacity expansion," Technique Report of Ashikaga Inst. of Tech., 1995.
- [11] D. E. Goldberg, 'Genetic Algorithms in Search, Optimization and Machine Learning,' Addison Wesley, 1989.
- [12] J. Lee and G. E. Johnson, "Optimal Tolerance Allotment Using a Genetic Algorithm and Truncated Monte Carlo Simulation," Computer-Aided Design, Vol.25, pp.601-611, 1993.
- [13] H. Luss, "Operations research and capacity expansion problems : a survey," Operations Research, pp.907-947, 1982.
- [14] A. S. Manne, 'Investment for Capacity Expansion : Size, Location and Time phasing,' MIT Press, Cambridge, 1967.
- [15] Z. Michalewicz, 'Genetic Algorithms + Data Structures = Evolution Programs,' Second Edition, New York: Spring-Verlag, 1994.
- [16] T. Odanaka, 'Dynamic Management Decision and Stochastic Control Processes,' World Scientific, 1990.
- [17] M. Gen and R. Cheng, 'Genetic Algorithms and Engineering Design,' John Wiley & Sons, Inc., 1997.

### 조 정 복

e-mail : jobok@kowon.dongseo.ac.kr

1978년 경북대학교 전자공학과 졸업(공학사)

1986년 영남대학교 대학원 전자공학과 전자계산기전공 졸업(공학석사)

1996년 동경도립과학기술대학 대학원 정보제어시스템 공학전공 졸업(공학박사)

1980년~1992년 경남전문대학 전자계산과 부교수

1991년~1992년 군마대학 공학부 정보공학과 연구자

1993년~현재 동서대학교 정보시스템공학부 컴퓨터공학전공 조교수

관심분야 : 퍼지이론, 유전자알고리즘, 데이터네트워크

### 양 황 규

e-mail : hkyang88@kowon.dongseo.ac.kr

1988년 경북대학교 전자공학과 전자계산학전공 졸업(공학사)

1990년 경북대학교 대학원 전자계산기공학과 졸업(공학석사)

1997년~현재 부산대학교 대학원 전자계산학과 박사과정 중

1990년~1995년 국방과학연구소 연구원

1996년~현재 동서대학교 정보시스템공학부 컴퓨터공학전공 전임강사

관심분야 : 신경회로망, 유전자알고리즘, 패턴인식

### 이 재 욱

e-mail : julee@kowon.dongseo.ac.kr

1975년 경북대학교 전자공학과 졸업(공학사)

1981년 동아대학교 대학원 전자공학과 졸업(공학석사)

1989년 동아대학교 대학원 전자공학과 졸업(공학박사)

1987년~1988년 아시카가공대 교환교수

1992년~현재 동서대학교 정보시스템공학부 컴퓨터공학전공 부교수

관심분야 : 퍼지이론, 유전자알고리즘, 시스템신뢰성

### 함 창 현

e-mail : hch77@nengjung.kit.ac.kr

1985년 경북대학교 전자공학과 전자계산학전공 졸업(공학사)

1989년 경북대학교 대학원 전자공학과 졸업(공학석사)

1996년~현재 경북대학교 대학원 컴퓨터공학과 박사과정 중

1992년~현재 경남정보대학 컴퓨터정보학부 전자계산과 조교수

관심분야 : 신경회로망, 유전자알고리즘, 패턴인식



## Baoding Liu

e-mail : liu@midwest.com.cn

1989년 Chinese Academy of Sciences, Institute of System Science(M.S.)

1993년 Chinese Academy of Sciences, Institute of System Science(Ph.D.)

1993년~1994년 Visiting Professor at Ashikaga Institute of Technology

1995년~현재 Department of Applied Mathematics, School of Science, Tsinghua University, Professor

관심분야 : Stochastic program, Fuzzy programming, Genetic algorithms