Thesis for the Degree of Doctor of Philosophy

Pareto Artificial Life Optimization Algorithms for Multi-objective Optimization Problems

by

JinDae Song

Department of Mechanical Engineering

The Graduate School

Pukyong National University

February 2010

Pareto Artificial Life Optimization Algorithms for Multi-objective Optimization Problems

다목적 최적화 문제를 위한 파레토 인공생명 최적화 알고리듬

Advisor: Prof. Bo-Suk Yang

by

JinDae Song

A thesis submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in Department of Mechanical Engineering, The Graduate School
Pukyong National University

February 2010

Pareto Artificial Life Optimization Algorithm for Multi-objective Optimization Problems

A Dissertation

by

JinDae Song

Approved by:

_____

Chairman: Prof. Soo-Jong Lee

_____ _____

Member: Prof. Byung-Tak Kim Member: Prof. Seon-Jin Kim

_____ _____

Member: Dr. Seong-Pil Choi Member: Prof. Bo-Suk Yang

December 18, 2009

Contents

List of Figures

List of Tables

Nomenclatures

G	number of current generation
$G_{\max}$	the maximum number of generations
α	factor to reduce the radius of proximity domain
C	neighbor region
N_{in}	number of artificial organisms
I_e	initial internal energy
N_{sp}	number of artificial species
G_e	generated energy during metabolism
E_e	elite energy
R_a	minimum age for reproduction
R_e	minimum energy for reproduction
R_p	probability of reproduction
L_e	losing internal energy
L_i	minimum energy for surviving
V_k	kth variation of colony
d_{ij}	Euclid distance of two objects, i and j
$f_{share(i)}$	shared fitness
d_j	density of solutions of object j (or candidate solution)
RC_0	radius to calculate density d_j on the basis of the location of object j
RC_j	radius of the final proximity
NAC_j	number of objects in proximity
W	bearing load
n_s	operating speed
ε_0	eccentricity ratio
p	oil film pressure
T	oil film temperature
F_j	journal surface friction
Q	supply flow

ω_{cr} whirling onset speed

λ diameter ratio

Pareto Artificial Life Optimization Algorithm for Multi-objective Optimization Problems

JinDae Song

Department of Mechanical Engineering, The Graduate School,
Pukyong National University

Abstract

The design of complex machinery is an iterative process. Components are redesigned iteratively until acceptable performance and reliability is obtained. In preliminary design of machinery, the optimum design is carried out to reduce the iterative redesign process. Therefore, the optimization of the system has become an important part of design process. There are very efficient optimization methods called the hill-climbing methods. However, it is clear that these methods provide local optimum values only and these values depend on the selection of the starting point. Therefore this research concerns with global optimization methods to overcome these problems. One of the global optimization methods is the artificial life algorithm (ALA) for function optimization. This study proposed a hybrid ALA called the enhanced artificial life optimization algorithm (EALA) to overcome the demerits of the ALA which are low speed of convergence and low accuracy after generating colony.

Most of engineering optimization problems often consists of several objective functions rather than a single objective function. Basically, there are two kinds of approaches to solve the multi-objective optimization problems (MOP). The first approach transforms a given multi-objective optimization problem into a single objective optimization problem (SOP). In order to provide possible solutions for the final decision maker, this approach has limitation that is only one solution is provided. The second approach is based on the concept of Pareto optimality to

avoid this difficulty and to explore various possibilities.

In order to apply artificial life algorithm to MOP in engineering problems, it is necessary to solve the Pareto optimization problem. Therefore, in this study, artificial life optimization algorithm has been expanded to enable the application of Pareto optimization to solve the MOPs.

I. Introduction

1. Background

The design of complex machinery is an iterative process. Beginning with concepts and configurations synthesized from experience, the preliminary design if analyzed with mathematical models using the best parametric data available. At every point where it is economically feasible, the data are verified or improved by testing. Components are redesigned iteratively until acceptable performance and reliability is obtained. Therefore, in preliminary design of machinery, the optimum design is carried out to reduce the iterative redesign process. Therefore, the optimization of the system has become an important part of design process.

There are very efficient optimization methods called the hill-climbing methods [1-3]. Hill-climbing methods use the iterative improvement technique which is applied to a single point (the current point) in the search space. During a single iteration, a new point is selected based on gradient of an objective function. It is clear that these methods provide local optimum values only and these values depend on the selection of the starting point.

Many researchers have proposed and developed various global optimization methods to overcome these problems. The best-known algorithms in this class include genetic algorithm (GA) [4-6], evolution strategies [7], ant colony [8], and artificial life algorithm [9, 10]. These algorithms are appealing to many uses in different areas of engineering, computer science, and operation research because of their simplicity, ease of interfacing, and extensibility.

2. Optimization

Designer, in industrial fields, supplies design parameters for the product as input into the computer simulation programs which is developed by commercial vender, runs the program and then analyzes the results. If the results do not meet the design goals then the designer changes the design parameters and repeats the process. Solutions to their problems have been based mostly on judgment and experience. However, increased competitions and consumer demands often require that the solutions should be optimum and not just feasible solutions. The challenge to the designer is to find the best design. It can be realized to the designer through the optimization.

Optimization is the process of maximizing or minimizing a desired objective function, which may be performance or weight, while satisfying the prevailing constraints.

This chapter introduces the general concept of optimization. The definition and history of optimization are considered from the view point of engineering.

2.1 Historical Review

The existence of optimization methods can be traced to the days of Newton, Lagrange and Cauchy. The development of differential calculus methods of optimization was possible because of the contributions of Newton and Leibnitz to calculus. The use of a gradient method (requiring derivatives of the function) for minimization was first presented by Cauchy in 1847. He made the first application of the steepest descent method to solve unconstrained minimization problems. In spite of these early contributions, very little progress was made until the middle of the twentieth, when high-speed digital computers made the implementation of the optimization procedures possible and stimulated further research on new methods. Modern optimization methods were pioneered by Courant's on penalty functions

in 1943. Dantzig developed the simplex method for linear programming in 1947 and Bellman stated the principle of optimal policy for system optimization for dynamic programming problems paved the way for development of the methods of constrained optimization in 1939. Kuhn and Tucker derived the “KKT (Karush, Kuhn and Tucker)” optimality conditions for constrained problems which laid the foundations of a great deal of later research in non-linear programming in 1951. Fletcher and Reeves proposed the conjugate gradient method that is pioneer on unconstrained minimization. Constrained optimization methods were pioneered by Rosen’s gradient projection method and Fiacco and McCormick’s SUMT (sequential unconstrained minimization techniques) techniques in 1968.

Geometric programming was developed by Duffin, Zener and Peterson. Gomory have done a pioneering work in integer programming, which is one of the most exciting and rapidly developing areas of optimization. Dantzig, Charnes and Cooper developed stochastic programming techniques by assuming design parameters to be independent and normally distributed. In the 1960’s, also, there were developments in non-gradient or direct methods such as principally Rosenbrock’s method of orthogonal directions in 1960, the pattern search method of Hooke and Jeeves in 1961, Powell’s method of conjugate directions in 1964, and the simplex method of Nelder and Meade [2]. Sequential quadratic programming (SQP) methods for constrained minimization were developed in the 1970’s. Development of interior methods for linear programming was started by the work of Karmarkar in 1984. Most approaches among direct methods in recent research were genetic algorithms (Holland [4], Goldberg [5]). Tabu search algorithm which was developed independently by Glover [11, 12] and Hansen [13] for solving combinatorial optimization problems and simulated annealing algorithms was derived from an analogy with the annealing process of material physics by Kirkpatric [14]. Special methods that exploit some particular structure of a problem were also developed. Pareto optimality was developed in the context

of multi-objective optimization. The use of nonlinear optimization techniques in structural design was pioneered by Schmit in 1960. Today, applications are everywhere, from identifying structures of protein molecules to decreasing the heat generation of journal bearing.

2.2 Definition of Optimization Problem

The design optimization problems are commonly found in manufacturing industries and can be represented by the following mathematically formulation.

Find $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in R^n$

which minimize $f(\mathbf{x})$

subject to the constraints $g_j(\mathbf{x}) \le 0,\ h_j(\mathbf{x}) = 0,\ j = 1 \text{ to } m$

where n is the dimension of variables and m is the total number of the constraint condition (or function). $\mathbf{x}$ is a real numbered vector of n dimension. $f(\mathbf{x})$ is an objective function or a cost function. $g_j(\mathbf{x}) \le 0$ and $h_j(\mathbf{x}) = 0$ are an inequality constraint and an equality constraint, respectively. If $\mathbf{x}$ satisfies $g_j(\mathbf{x}) \le 0$ and $h_j(\mathbf{x}) = 0$, an $\mathbf{x}$ is called a feasible solution.

This formulation supports the specification of unconstrained and constrained problems with a single objective. In the optimization problem formulation, three elements are considered such as design variables, constraints and an objective function.

The idea of improving or optimizing a machine implicitly presupposes some freedom to change the machine. The potential for change is typically expressed in terms of ranges of permissible changes of a group. Such parameters are usually called design variables and denoted by a vector $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in R^n$ .The choice of design variables can be critical to the success of the optimization process. In particular it is important to make sure that the choice of design variables is consistent with the analysis model.

Constraints introduce the notion of limits on the design variables in the optimization problem formulation. Because of their simplicity, these upper and lower limit constraints on the values of the design variables are often treated in a special way by solution procedures, and are refereed to as side constraints. Constraints which impose upper or lower limits on quantities are by their very some strategies for the solution of nonlinear optimization problems are unable to handle equality constraints, but are limited to inequality constraints only. In such instances it is possible to replace the equality constraint with two inequality constraints that form upper and lower bound constraints with a same limiting value. However, it is usually undesirable to increase the number of constraints.

The objective function, when expressed as a function of the design variables, is known to the criterion with respect to which the design is optimized. The choice of objective function is governed by the nature of the problem. For mechanical engineering problems, weight, displacement, stresses, vibration frequencies, buckling loads, increase of temperature, flow rate of lubricant, and cost or any combination of these can be used as objective functions.

3. Artificial Life

Artificial life (AL) is defined as the attempt to study all biological phenomena of the living world by reproducing them in artificial systems [9, 15, 16]. This ordinarily means to simulate living phenomena in a computer, and is a man-made life system whose characteristic behaviors include self-reproduction, self-organization, evolution, and so on [17]. The research motive of AL was originated from the intent to understand the true meaning of life through the synthesis of the life that makes it superior to the existing life in nature. They have succeeded in generating creatures that look and act very many like living organisms on the computer screen. They can grow, reproduce, mutate, fight with each other, and die out [18].

The most important characteristics of AL are emergence and dynamic interaction with the environment. Namely, the micro-interaction with each other in the AL's group results in emergent colonization, the emergence, in the whole system. The concrete study method using the above characteristics consists of mainly two steps. First, the essence of AL system, which shows the behavior characteristics of living organisms in the natural world such as growth, adaptation, multiplication, self-preserving, self-control, and evolution, is realized through several theoretical models. Second, the AL organisms which are called real living organisms are created in the computer through the simulation. This process can be defined as the informationization process. Therefore, the research object of AL is not the physical system of life itself but the function as the information. AL has been associated to computer science by different ways. Many researchers have been involved with this field looking for models that would describe how real life began and evolved [19]. Those system models using artificial life's characteristics can be classified into as follows:

- Adaptive moving model: classifier system [20], neural network [21], immune system [22]
- Evolution operation model: genetic algorithm [23,5], evolutionary strategy [24], evolutionary programming [25], genetic programming [26], co-evolution [27]
- Genetic development model: L-system [28], cellular automata [29-31]

Although very different in nature, these works have been related to evolution and natural selection concepts.

Many heuristic methods currently used in combinational optimization are inspired by adaptive natural behaviors or natural systems, such as genetic algorithms (GAs), simulated annealing, neural networks, ant colony algorithms, etc. The GAs is computational procedures to find the optimal solution in particular hard problems [22, 32]. This strategy provides an efficient way to obtain global optimization in cases where it is very difficult or not practical to formalize the optimization problem on an analytical framework. The ant colony algorithms are based on the principle that using very simple communication mechanisms, an ant group is able to find the shortest path between any two points [8].

4. Summary of This Dissertation

The dissertation is briefly summarized as follows.

Chapter II presents an approach by hybrid artificial life algorithm with random tabu search method so called the enhanced artificial life optimization algorithm (EALA) for function optimization problem. The effectiveness of the proposed algorithm is evaluated using three test functions. The optimized results were compared with those of conventional artificial life algorithm for function optimization.

Chapter III proposes a modified artificial life algorithm namely the Pareto artificial life optimization algorithm (PAL) to handle a multi-objective optimization problem. Through three known test problems, searching capabilities of the PAL are verified. Also, set of optimum solutions is found out by applying two objective functions to the optimum design of the journal bearing. By comparing with the optimum solutions of single objective function, it is confirmed in examples of this study that the result of single objective function optimization is one result of the specific cases of multi-objective optimization.

Chapter VII summarizes and discusses the results obtained this paper.

References

[1] S.S. Rao, 1996, Engineering Optimization, John Wiley & Sons, Inc.

[2] J.A. Nelder, R. Mead, 1965, “A Simplex Method for Function Minimization,” Computer Journal, 7, 308-313.

[3] M.J. Box, 1965, “A New Method of Constrained Optimization and a Comparison with Other Methods,” Computer Journal, 8(1), 42 -52

[4] J.H. Holland, 1975, Adaptation in Natural and Artificial System, the University of Michigan Press, Michigan.

[5] D.E. Goldberg, 1989, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Co. Inc., New York.

[6] L. Davis, Ed., 1991, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, USA.

[7] H.P. Schwefel, 1981, Numerical Optimization fir Computer Models, John, Wiley & Sons, Inc., UK.

[8] M. Dorigo, V. Maniezzo, A. Volorni, 1996. “The Ant System: Optimization by a Colony of Cooperating Agents,” IEEE Trans. Systems, Mans, and Cybernetics (Part B), 26, 29-41.

[9] C.G. Langton, Ed., 1989. Artificial Life, Addison-Wesley Publishing Company.

[10] B.S. Yang, Y.H. Lee, 2000, “Artificial Life Algorithm for Function Optimization,” ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Baltimore, DETC 2000/DAC-14524.

[11] F. Glover, 1986, “Future Paths for Integer Programming and Links to Artificial Intelligence, “ Computers Operations Research, 5, 533-549

[12] F. Glover, 1989, “Tabu Search- Part I,” ORSA Journal on Computing, 1(3), 190-206

[13] P. Hansen, 1986, “The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming,” Numerical Methods in Combinatorial Optimization, Capri, Italy.

[14] S. Kirkpatrick, C. Gelatt, M. Vecchi, 1983, “Optimization by Simulated Annealing,” Science, 220(4598), 671-690.

[15] C.G. Langton, Ed., 1995, “Artificial Life: An Overview,” MIT Press, MA.

[16] A.M. Assad, N.H. Packard, 1992. Emergent Colonization in Artificial Ecology, Technical Report CCSR-92-3.

[17] D. Parisi, M. Schlesinger, 2002, “Artificial Life and Piaget,” Cognitive Development, 17, 1301-1321.

[18] J. Hu, H. Hu, Y. Fang, 1996, “Research into Models and Algorithms of Artificial Life,” Artificial Intelligence in Engineering, 10, 299-310.

[19] C. Adami, 1998, Introduction to Artificial Life, Springer, Berlin.

[20] S. Forrest, J.H. Miller, 1990, “Emergent Behavior in Classifier Systems,” Physica D, 42, 213-227.

[21] S. Rasmussen, H. Karampurwala, R. Vaidyanath, L.S. Jensen, S. Hameroff, 1990, “Computational Connectionism within Neurons: a Model of Cytoskeletal Automata Subserving Neural Networks,” Physica D, 42, 428-449.

[22] A. Somayji, S. Hofmeyr, S. Forrest, 1998, Principles of a Computer Immune System, New Security Paradigms Workshop, Cumbria, UK.

[23] J. Holland, 1992. Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA.

[24] H. Schwefel, 1995, Evolution and Optimum Seeking, Wiley, New York.

[25] L. Fogel, A. Owens, M. Walsh, 1966, Artificial Intelligence through Simulated Evolution, Wiley, New York.

[26] J.R. Koze, 1992, Genetic Programming, MIT Press, Cambridge, MA.

[27] S.A. Kauffman, S. Johnsen, 1992, “Co-Evolution to the Edge of Chaos: Coupled Fitness Landscapes, Poised States, and Co-Evolutionary

Avalanches," Artificial Life Ⅱ: Volume Ⅹ of SFI Studies in the Sciences of Complexity, Addison-Wesley Publishing Company.

[28] A. Lindenmayer, P. Prusinkiewicz, 1989, "Developmental Models of Multicellular Organisms: A Computer Graphics Perspective," Artificial Life: Volume Ⅵ of SFI Studies in the Sciences of Complexity, Addison-Wesley.

[29] J. Von Neumann, 1966, Theory of Self-Reproducing Automata, University of Illinois Press, IL.

[30] S. Wolfram, 1986, Theory and Applications of Cellular Automata, World Scientific, Singapore.

[31] M. Sipper, 1995, "Studying Artificial Life Using a Simple, General Cellular Model," Artificial Life, 2(1), 1-35.

[32] M. Mitchell, 1997, An Introduction to Genetic Algorithms, MIT Press, MA.


PUKYONG NATIONAL UNIVERSITY
부경대학교

Ⅱ. Enhanced Artificial Optimization Algorithm

1. Introduction

Recently, the possibilities of using ALA in optimization applications were being considered only by Satoh et al. [1], Yang et al. [2, 3] and Yang et al. [4]. Its development as a standard optimization tool is still in its infancy. Many aspects require considerable research effort. In ALA for the function optimization, the emergent colonization is accomplished through the metabolism and the reproduction in the artificial world. The optimum solutions are found on the emergently colonized region [2].

Existing mathematical programming methods for the differentiable irregular function such as the conjugate gradient method, sequential quadratic programming approach among others [5], make use of local curvature information derived from linearization of the original functions by using their derivatives with respect to the design variables. These methods present a satisfactory local rate of convergence, but they cannot assure that the global optimum can be found and have the risk of trapping in the local optimums because those methods have an extreme dependence on the initial value. Tabu search method is a metaheuristic strategy for solving global optimization problems [6, 7]. This is called a metaheuristic because it can be combined with other heuristic procedures to prevent them from trapping at locally optimal solutions. It is an iterative procedure that starts from some initial feasible solution and attempts to determine a better solution. Tabu search method, basically a kind of neighborhood search method, makes several neighborhood moves and selects the move producing the best solution among all candidate moves for current iteration. This best candidate solution may not improve the current solution. Selecting the best move is based on the supposition

that good moves are more likely to reach the optimal or near-optimal solutions [8]. Hu [9] proposed a tabu search method with random moves for global optimization of continuous variables.

Yang et al. [2, 3] proposed an artificial life algorithm (ALA) that can be applied to the irregular function without depending on the initial value [2] and also applied it to the optimization of the short journal bearing [2] and engine mount [10] to verify it. In the ALA, the emergent colony is the fundamental mechanism to search the optimum solution. Emergent colonies being accomplished through the metabolism, movement and reproduction among artificial organisms appeared at the optimum locations in the artificial world. The locations are optimum solutions in the optimization problem. Then, the ALA focuses on the searching of the optimum solution in the location of emergent colonies and can find a more accurate global optimum. So, the ALA is a stochastic searching algorithm using the feature of artificial life. But, the ALA has a demerit that after it is congregated at the neighborhood of optimum solutions, not only does the convergent speed become very slow, but also the solution accuracy is not good. Moreover, to decide the location of the waste in metabolism, of random movement, if the individuals did not find the wanted resources is difficult. In the next generation, the reproduction having a very important influence to the efficiency of the ALA remained to be improved [11].

This study proposes a hybrid ALA called the EALA (Enhanced Artificial Life optimization Algorithm) to overcome the above mentioned problems. In this section, the parameters used in the EALA are studied to simplify the parameters through normalizing and relating parameters. Also the performance of the EALA is verified.

2. Artificial Life Algorithm (ALA)

The artificial world is defined as the space where the lowest limit and the highest limit of the variables are $x_i^{\min}, x_i^{\max} \in R^n \left(i = 1, 2, \cdots, n\right)$, respectively. For example, it is supposed that there are four kinds of resources (white, red, green and blue) and four species of artificial organisms (white, red, green and blue) in two dimensional space (n = 2).

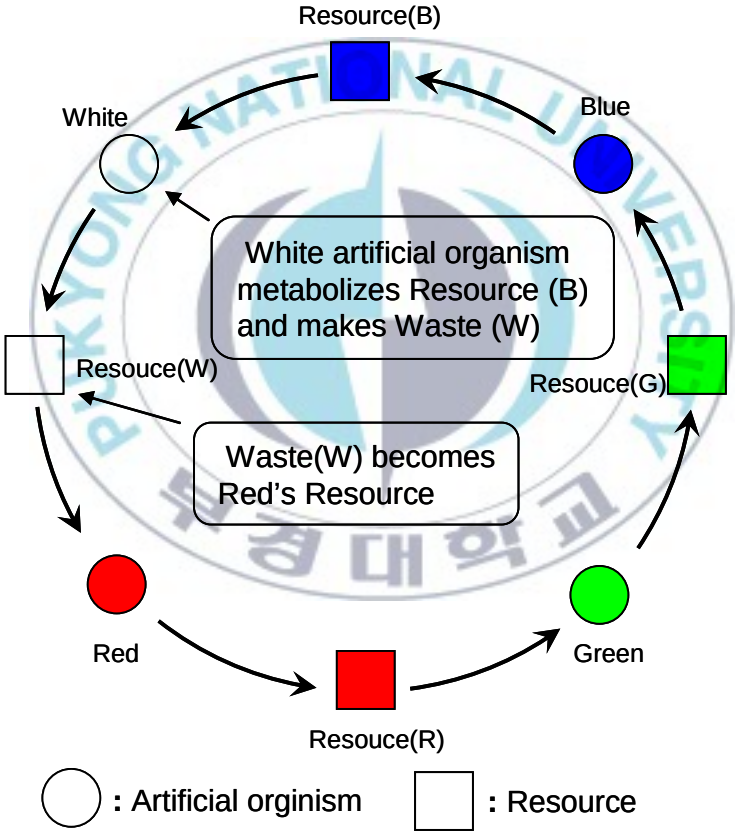


Fig. 2.1 A Circular Food Chain of an Artificial Organism.

Artificial organisms can move about in the world consuming energy resources and producing waste. The four species of artificial organisms compose a circular food chain where one species' waste is another's food as shown in Fig. 2.1. Artificial organisms can only metabolize the resources they want. The demanded resources are determined according to the four species of artificial organisms. When they metabolize resources, their internal energy is increased by increasing internal energy G_e when they eat the wanted resources. After metabolizing it, they produce the waste at the random location of their neighborhood region. This waste becomes the resource for another artificial organism. A white artificial organism metabolizes a blue resource which it wants and produces waste. Then this waste becomes a white resource which is then metabolized by red artificial organism. This relation is carried out among the four species of artificial organisms.

Artificial organisms have a sensory system which enables them to search resources as well as other artificial organisms in the world. They are also able to determine the location of the nearest resources and other artificial organisms from their present location. This nearest location of resource becomes the goal which drives them to move forward. Artificial organisms must maintain a minimum internal energy level L_i in order to survive. Once an artificial organism's energy level drops below L_i, it is considered to be dead and removed from the world. Whenever artificial organism's age is increased by 1, its internal energy is decreased by L_e. Therefore, although artificial organisms do not have a pre-specified lifespan, they are subject to the consuming rate of energy which is proportional to their age. Thus, as an artificial organism grows older, it should be incessantly supplied with energy to maintain its existence. Eventually, if it doesn't have enough chance to metabolize resources and get energy, it is supposed to die. This condition effectively imposes a finite lifespan on artificial organisms. Fig. 2.2 shows the definition of a neighborhood region using two variables problems

as an example. The location $\mathbf{x}_s$ of an arbitrary artificial organism (individual) is considered as the center. Based on this point, the neighborhood region C of $\mathbf{x}_s$ is the space within the Euclidean distance and C is defined as

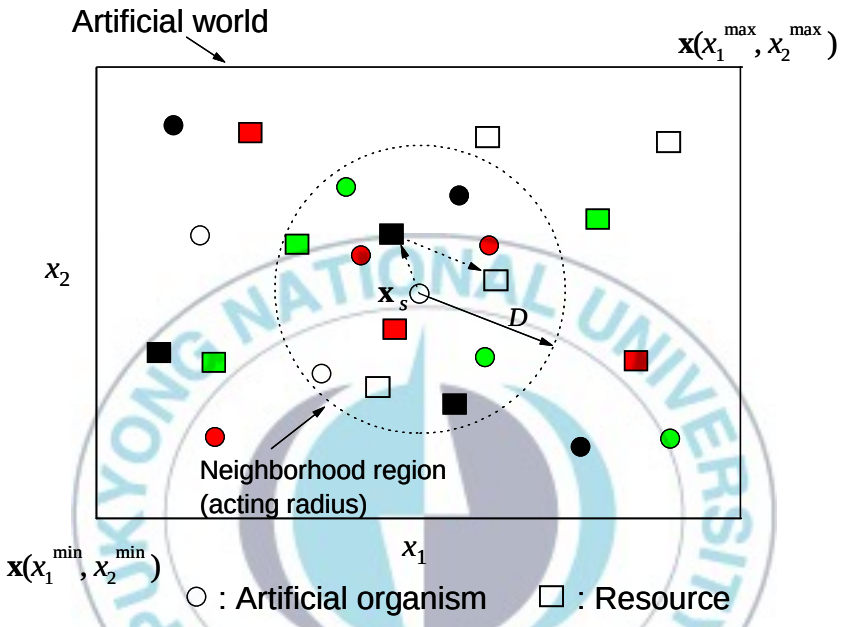


Fig. 2.2 Definition of a Neighborhood Region.

$$C = \left\{ \mathbf{x} \in R^n \,\middle|\, \left\| \mathbf{x} - \mathbf{x}_s \right\| \le D \right\} \tag{2.1}$$

where n is the dimension of the artificial world and/or the number of design variables, D is the possible movement range of artificial organisms per generation given by,

$$D = D_0 \cdot e^{-(t/T)\alpha} \tag{2.2}$$

where D_0 is the initial value, t is the generation number, T is the last generation

number, and α is the radius parameter.

Artificial organisms use this defined neighborhood region for their movement and reproduction. They can only see and find the resources and other organisms within this region for each generation. *D* can be considered as a constant for the whole generation or as a variable according to increasing generation. In this study, artificial organisms can only move about and find other artificial organisms within the neighborhood region defined above per generation. If there are no resources in the neighborhood region, they can move randomly within the neighborhood region.

In case of the optimization problem, the artificial world becomes the space of the design variables **x**. Every location has its own fitness. After they search the neighborhood region randomly, they produce their offspring finally at the location which has higher fitness than those of themselves.

Also, the sensory system is introduced to enable artificial organisms to see and find the optimum value in the artificial world. These resources become the goal which drives an artificial organism to move towards a destination. Thus, all artificial organisms move to the resources which have the optimum value. Eventually, it is more likely to produce an emergent colonization at the location of the resources in the artificial world.

The location of both artificial organisms and resources becomes the variables of the objective function which to optimize. Therefore, the objective function values for artificial organisms and resources can get by substituting the location into the objective function. An artificial organism compares its objective function value to that of the resources within the neighborhood region. It moves to the location of the resource having the higher fitness within the neighborhood region. Therefore, artificial organisms can produce an emergent colonization at the location that has the optimum of objective functions.

3. Tabu Search Method with Random Moves

In this study, the moves in neighbors of different sizes are used both to prevent it from trapping at a local optimum as in hill climbing based methods, and to avoid blindly sampling as in random search [9]. The random tabu search method was recently extended for function optimization in the domain of continuous variables with the general constraints [12]. In the random tabu method, a set of steps h_1, $\mathbf{H} = \{h_1, h_2, \cdots, h_{N_s}\}$ is given. For an initial feasible point x, the search moves are made over a set of active steps l, $l = \{1, 2, \cdots, N_s\}$, where the step $h_1 \in \mathbf{H} - \mathbf{T}$ and **T** is the tabu list of accepted steps, which is initially empty. For each active step one feasible random move is generated as $x_i^{k+1} = x_i^k + r \cdot h_{i,l}$, where r is a random number $(-1 \le r \le 1)$. If there is a decrement in the objective function, the random move is saved as the current solution **x** and the step h_1 is added to **T**. When **H** – **T** is empty, **T** is updated empty and the total process is repeated, otherwise the above procedure is repeated. As mentioned by Hu [9], there is a probability for each point of the search space which is exploited. In this procedure, the moves in neighbors $\left(x_i^k - h_i \le x_i^{k+1} \le x_i^k + h_i\right)$ of different sizes prevent it from trapping at a local optimum.

The procedure used to calculate H is:

First, assume that the objective function is defined as a function of n continuous variables x_i, $i \in \{1, \cdots, n\}$ in a box **R**,

$$\mathbf{R} = \{\mathbf{x} \mid a_i \le x_i \le b_i, \cdots, a_n \le x_n \le b_n\} \tag{2.3}$$

Then, the N_s steps of **H** can be calculated as follows:

$h_{i,1} = (b_i - a_i) / c$

$h_{i,2} = h_{i,1} / c$

…

$h_{i,Ns} = h_{i,Ns-1} / c$

where c is a factor greater than 1, for example $c = 2$.

Features of the random tabu method are as follows. First, it can reduce the number of iteration and promote the efficiency of searching, because each searching solution locates at different searching domain. Second, this method is possible to take a global optimum and to avoid trapping in local optimum because of utilizing random searching. Finally, it is possible to get the optimum solution fast and accurately, if the method is combined with other global optimization methods.

4. Enhanced Artificial Life optimization Algorithm (EALA)

In the ALA [2], how fast the colonies at the location of the optimum solution can be made is one of the most important factors determining the performance of optimization. Also, the amount of individual density that the individuals located in the colonies had for the area of the colonies is an important factor determining the accuracy of the solution. They do not only determine the efficiency of the concentrate search but also mean the level of high solution accuracy. The timing of colonies formation and the individual density basically depend on the following three decisions of location problem. The bottom line is how to decide the new location efficiently.

- How to decide the location of the waste in metabolism
- How to decide the location of random movement, if the individuals did not find the wanted resources
- How to decide the location of the next generation in reproducing

In the ALA, the possible searching space of each individual is defined by the neighbor region C as shown in Eqs. (2.1) and (2.2), in which organisms can perceive [2]. The ALA randomly searches the solution in this region. The ALA has the searching ability of the global optimum solution and the merit in searching speed is much better. But the individual density of colonies in the ALA is a little bit diverged as shown in Fig. 2.2. This means the efficiency of the precision search is low in the ALA.

In this study, it is possible to set the new locations discussed before at or near the sub-step of the optimum solution and to search more efficiently because the neighbor region is divided into several sub-steps by the RTS that was introduced into the ALA and then, to search the sub-steps. Thus, the colonization can be achieved quickly and precisely. Accordingly as artificial organisms are forming

the colonization in whose neighbor of the optimum solution point is included, the artificial organisms, which have the higher fitness and are near the location with the optimum solution, can search for the solution accurately in inner sub-steps having a small radius, so the total accuracy is increased. In addition, the outer part of the colonized group, which belongs to the colony, but relatively has more distance from the optimum point, also can select the new locations in the sub-step having the large radius and being nearest to optimum point. This makes it possible to produce a dense colonization. Therefore, the EALA has good features which are the converging speed is higher and the solution is more accurate than those in the ALA.

Fig. 2.3 shows the flow chart of the EALA and the calculation procedure of the EALA is as follows.

Step 1: The initialization is activated based on the initial determination. The number of artificial organisms and resources N_{in} are distributed randomly and the initial internal energy I_e is given to the each artificial organism. The number of individuals and resources for each species are obtained by dividing N_{in} by N_{sp}.

Step 2: Artificial organisms search for the nearest resource within their neighbor region as defined in Eq. (2.1).

Step 3: If they find the resource that they want to metabolize within their neighbor region, they move to it and metabolize it. During metabolism, they get the energy of G_e and produce waste at the random location in the neighbor region. It is assumed that during the metabolism, artificial organisms intake the resources and then, the resources are vanished. In the decision of the random location for waste, first, the neighbor region is divided into several sub-steps and then, the random point is selected in each sub-step to estimate the fitness of it for the comparison with the present fitness. If estimated

fitness is better than the present fitness, the selected location is determined as a candidate and if not, a new random point is selected. After the comparison of all candidates determined in all sub-steps, the point having the best fitness is determined as the new location of waste.

Step 4: If they did not find the wanted resource within their neighbor region, the random tabu method is applied to its moving method as in Step 3. The elites get as much energy as E_e.

Step 5: The age of artificial organisms and the number of the generation are increased by 1.

Step 6: The reproduction process starts. If the age and the internal energy of an artificial organism are not less than the minimum age for reproduction, R_a and minimum energy for reproduction, R_e respectively, the organism can mate according to the probability of reproduction, R_p with the closest artificial organism of the same species that satisfies the above two conditions for R_a and R_e. Two parent organisms reproduce themselves and the initial location of each offspring is determined within the neighbor region of each parent by the random tabu method.

Step 7: The internal energy makes a decrease as L_e and if the internal energy of an artificial organism drops below L_i, it is considered to be dead and is removed from the world.

Step 8: The number of generation is increased by 1. In returning to Step 2, the process is iterated until the final generation.

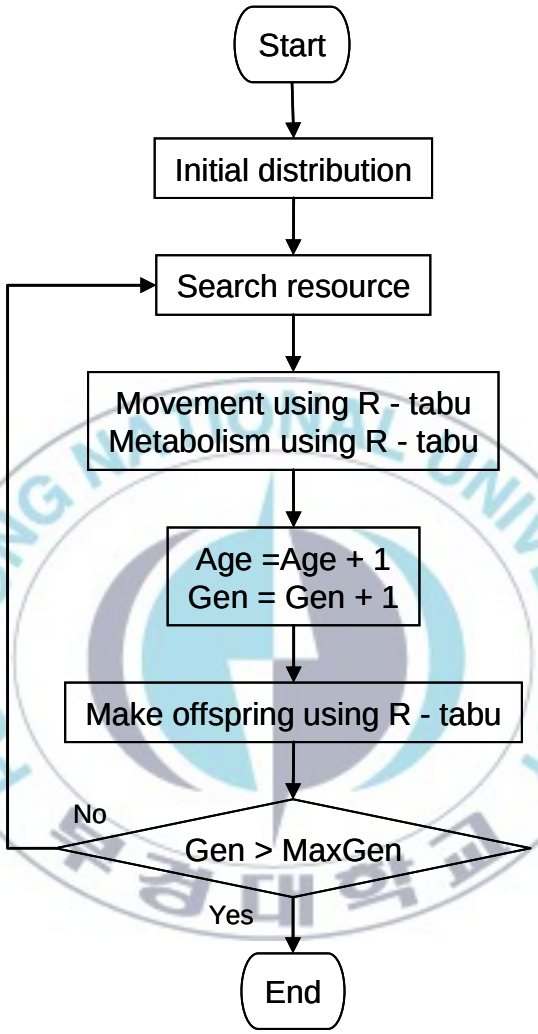


Fig. 2.3 Flow Chart of the Proposed Optimization Algorithm.

5. Estimation of Parameters of the EALA

Table 2.1 shows the parameters used in the EALA. These parameters can be divided by four categories as follows based on the major effect.

- The parameters related to individual energy for survival
- The parameters related to the number of individuals being survived
- The parameters related to searching
- The parameters concerned with initial conditions

Table 2.1 Parameters in the EALA

Symbol	Value	Symbol	Value	Symbol	Value
E_e	10	L_i	125	R_a	3
G_e	50	N_s	5	R_e	150
I_e	150	α	12	R_p	0.0002
L_e	5	P_e	1	N_c	3
R_r	10				

5.1 Parameters Related to Individual Energy

The parameters related to individual energy for survival are I_e, G_e and E_e, and are normalized by the minimum energy, L_i for survival and the consume energy, L_e according to the increasing age, as $L_i = 0$, I_e, G_e and E_e are normalized as an integer by L_e. The investigated regions to determine the value of I_e, G_e and E_e are $I_e = 3L_e \sim 20L_e$, $G_e = 3L_e \sim 30L_e$, and $E_e = 2L_e$, respectively. Four global optima of the multi-modal function are found in all values of L_e, G_e. Also, the calculation time and the accuracy of the solution with the variation of the I_e and G_e value do not change as the specific trend does. But it is identified that the number of expired individuals and produced individuals are reduced. The reduction of produced individuals means that the probability of searching by the new

individuals is reduced. Therefore, I_e = 5L_e, G_e = 5L_e means that the number of expired and produced individuals is comparatively high as selected in this study. Fig. 2.4 shows the total number of expired individuals for I_e = 3, 10 and 20.

In this study, the elite individuals can acquire double energy, E_e = 2L_e, as the intake energy to give the benefit of survival during generations while the intake energy of common individuals is just G_e = L_e.

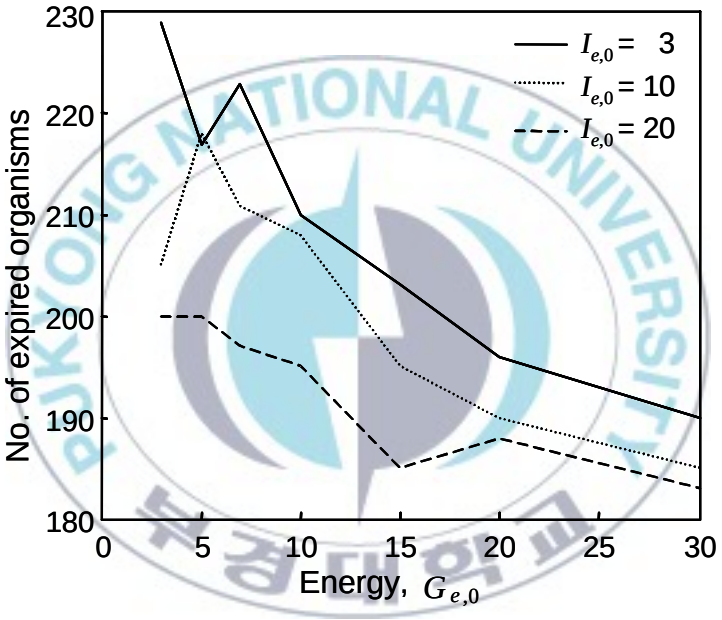


Fig. 2.4 Expired Organisms to I_e and G_e.

5.2 Parameters Related to the Number of Individuals

The parameters concerned with the convergence of individuals are decided by considering the normalized energy and the total number of individuals. The adult energy, the maximum energy at an adult age R_a, is defined in Eq. (2.4).

$$R_e = I_e + R_a \cdot G_e \tag{2.4}$$

The probability of reproduction, R_p, that determines whether the individuals reproduce them or not is controlled to keep the optimum individual. The number of optimum individual, N_{opt}, is the result of the addition of the total individuals that survived in the present generation, N_a, and the number of offspring, N_{off}, produced in the production. The number of offspring, N_{off}, can be obtained by multiplying the probability of finding a mate in the neighboring region, the ratio of production, R_p and N_a. As the probability of finding a mate in the neighbor region can be assumed approximately as 1, $N_{off} = R_p \cdot N_a$ is possible and so, the number of optimum individuals is $N_{opt} = N_a\left(1+R_p\right)$. Therefore, the ratio of production, R_p, is defined as follows;

$$\begin{cases} R_p = \dfrac{N_{opt} - N_a}{N_a}, & if\ R_p > R_{p,\min} \\ R_p = R_{p,\min}, & if\ R_p \le R_{p,\min} \end{cases} \tag{2.5}$$

where the minimum production ratio $R_{p,\min}$ has a maximum value when $N_a = N_{opt} - 1$, namely N_{off} = 1. Therefore, the ratio of production can be defined in Eq. (2.6).

$$R_p = \frac{N_{opt} - (N_{opt} - 1)}{N_{opt} - 1} = \frac{1}{N_{opt} - 1} \tag{2.6}$$

Also, the effect of N_{opt} and $R_{p,\min}$ to the number of individuals produced appears by examining N_{opt} and $R_{p,\min}$ at the same time and the recommended range considering the success of searching and the calculation time is as follows:

$$5.0\times10^{-7} \le R_{p,\min} \le 1/\left(N_{opt} - 1\right) \tag{2.7}$$

$$64 \le N_{opt} \le 240 \tag{2.8}$$

Fig. 2.5 shows the calculation time according to N_{opt} and $R_{p,\min}$. In the case of

both parameters having the value below the lower boundary of the recommended range, only three global optima are obtained occasionally. Namely, a few individuals can have the failure of searching and an excessive amount of individuals increase the calculation time.

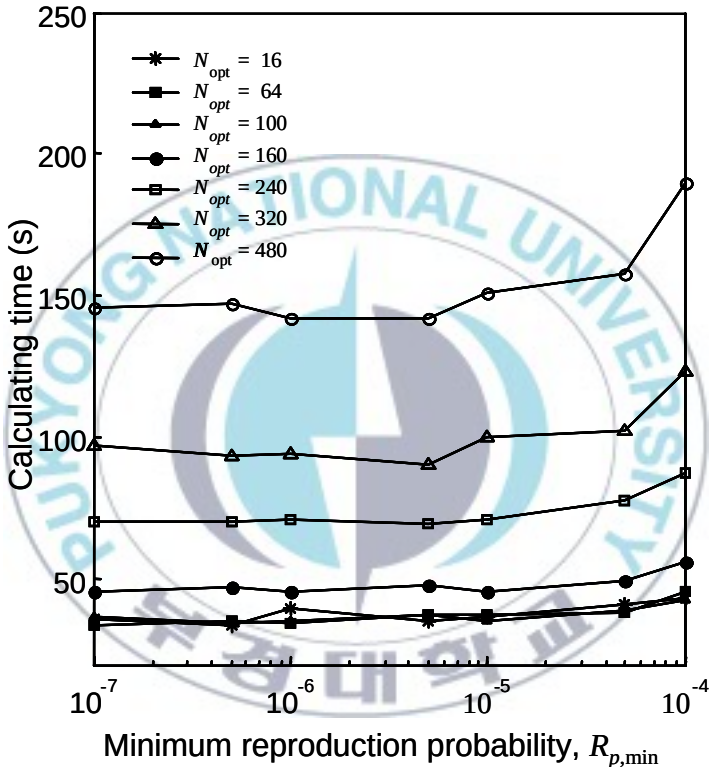


Fig. 2.5 Calculating Time According to N_{opt} and $R_{p,\min}$.

5.3 Parameters Related to Searching

The parameters directly related to searching are established by the following assumption. First, the neighbor region is decreased exponentially according to the increasing generation by Eqs. (2.1) and (2.2). Second, it is controlled so that the inmost sub-step of the last generation can reach the objective value, $R_{f,Ns}$, of the

optimum precision. Parameters α, N_c, N_s, $R_{f,Ns}$ and R_r must be determined. $R_{f,Ns}$ is directly concerned with the required precision of variables and determines the level of the solution that the designer requests in the design variable.

$$R_{f,Ns} = \frac{R_f}{R_r^{Ns-1}} \tag{2.9}$$

R_r is obtained by $R_{f,Ns}$ and is determined as follows.

$$R_r = \sqrt[N_s-1]{\frac{R_f}{R_{f,N_s}}} = \sqrt[N_s-1]{\frac{D_0 \exp(-\alpha)}{R_{f,N_s}}} \tag{2.10}$$

where R_f is the radius of the neighbor region at the final generation. Also, R_r >1 must be satisfied to reduce the radius of the inner sub-step. For D_0 = 1,

$$R_r = \sqrt[N_s-1]{\frac{D_0 \exp(-\alpha)}{R_{f,N_s}}} > 1 \tag{2.11}$$

Therefore,

$$0 < \alpha < -\ln\left(R_{f,Ns}\right) \tag{2.12}$$

The precision becomes worse as α increases. The increasing α, which makes the radius of the inner sub-step large, drops the effect of the precision search. The radius of the outermost and inmost sub-step is only determined by α and $R_{f,Ns}$. Therefore, the radius of each sub-step in the generation is the division of area, which is determined by α and $R_{f,Ns}$, into N_s using the linear-log coordinate (Fig. 6). Also, if $N_s = R_{f,Ns}$, the larger α makes R_r small, because the objective value $R_{f,Ns}$, is the same. Therefore, the radius of all the sub-steps except the inmost and outermost sub-step become much larger as the α gets larger in the generation before passing through the diagonal line in Fig. 6. The radius of the

sub-step according to the generation for α = 12 and α = 6 is also shown in Fig. 2.6. If N_s is increased, the calculation time is increased. The amount of calculation time obtained during 3000 generations according to increasing the number of the sub-steps is handled statistically through the total 36 cases of the numerical experiment in the case that N_s is 3, 5, 7, 10, 15 and 20, and α is taken as 3, 5, 7, 10, 12 and 15. As a result, the standard deviation $\sigma_{T/Ns}$ is 0.2 (unit time) for the mean $\Delta T / \Delta N_s = 1$ (unit time). Based on this result, it can be identified that increasing of the calculation time is approximately linear as the number of sub-steps increases. The trend of a result like that can be expected as the number of sub-steps increases. So, when N_s is determined, the calculation time must be considered. N_c is determined independently. The probability of success of searching by having just one random search in each step can be defined as

$$p = A_{better-fitness} / A_{sub-step} \tag{2.13}$$

where each region is a square area for the problem of two variables and cubic volume for the problem of three variables, and so on (Fig. 2.7). Fig. 2.7 shows the radius of the sub-step for an organism and contour lines.

The maximum probability, which can be succeeded at N_c time execution, follows the geometric distribution that is one of the discrete probability distribution functions. The maximum probability of success of searching is determined in Eq. (2.14).

$$P_{Ns} = \sum_{i=0}^{N_c-1} pq^i \tag{2.14}$$

where q is the probability of failure and p is the probability of success of searching at each execution. For example, if there is a contour of a circle for the optimum point and the boundary of the neighbor region of a specific individual pass through the optimum point as in Fig. 2.7, the probability, which this

individual can succeed in searching at the outermost sub-step, is approximately 0.33, 0.56, 0.70, 0.80 and 0.87 when N_c is 1, 2, 3, 4 and 5, respectively. Also, the probability of success is increased because p increased in the inner sub-step. Though N_c exceeds 3, the increase of probability of success is not larger than that of N_c. So, it can be known that N_c = 3 is the proper value for considering the calculation time.

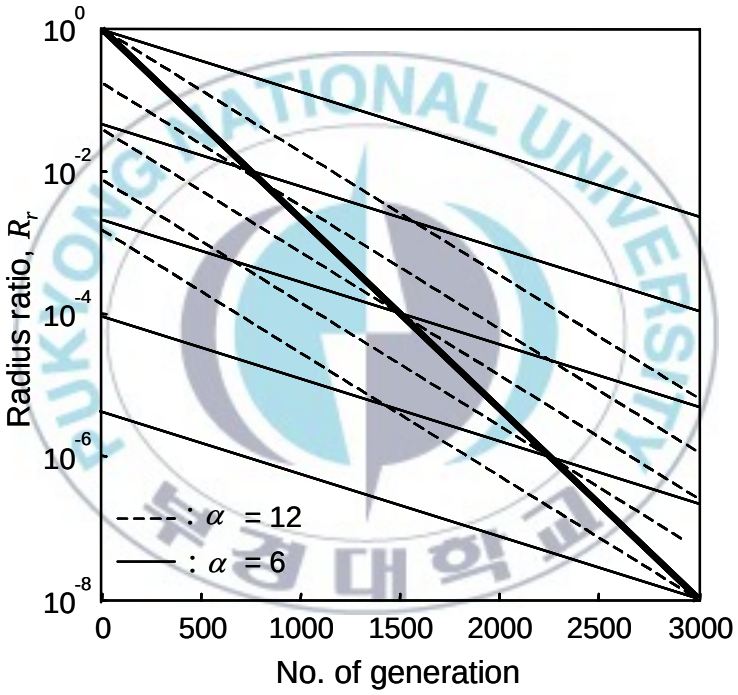


Fig. 2.6 Radius Ratio According to Generations.

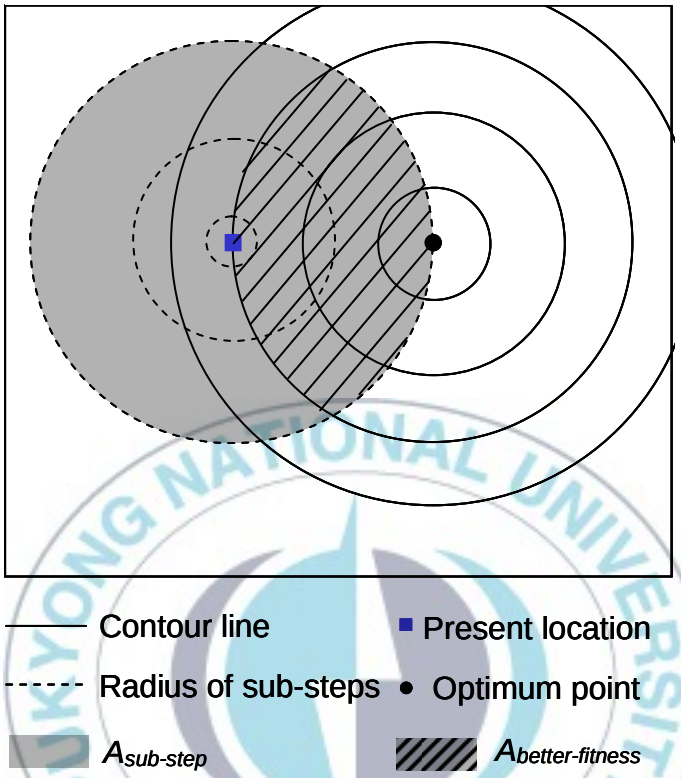


Fig. 2.7 The Radius of the Sub-step for an Organism and Contour Lines.

5.4 Parameters Concerned with Initial Conditions

It becomes clear from Fig. 2.8 there is few effects of the number of species among the parameters concerned with initial conditions. In order to compare the optimization performance, the calculation time and the approximate E_{rms} (root mean square) of objective function for all survived individuals through the generations are shown in Fig. 2.8. The E_{rms} is calculated by Eq. (2.15).

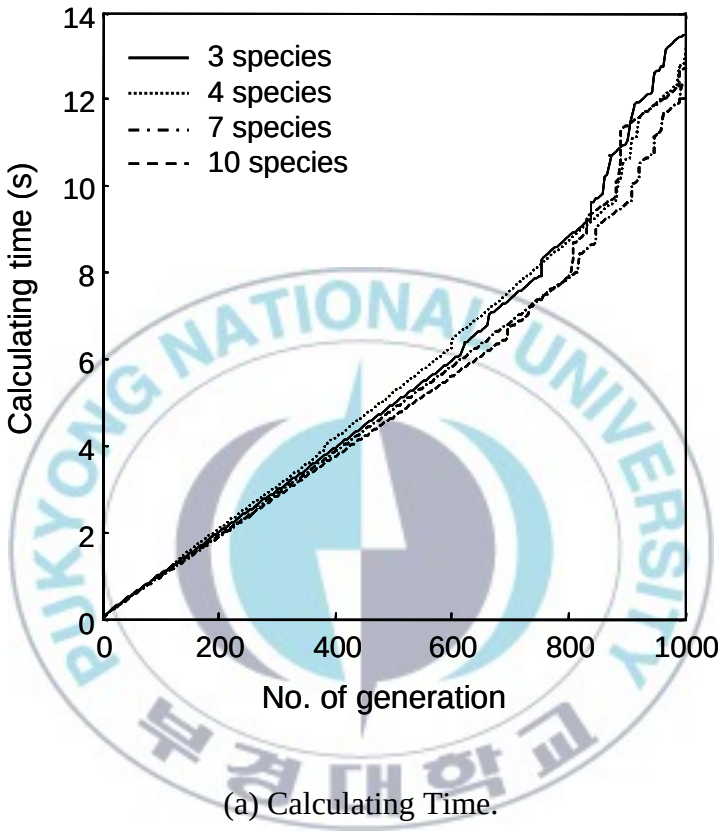
$$E_{rms} = \frac{1}{N}\sum_{j=1}^{M}\sum_{i=1}^{N_j}\left\| f_{j,opt} - f_{j,i} \right\| \tag{2.15}$$

where $N = N_1 + N_2 + \cdots + N_M$, N_j is the number of survived organisms in the jth

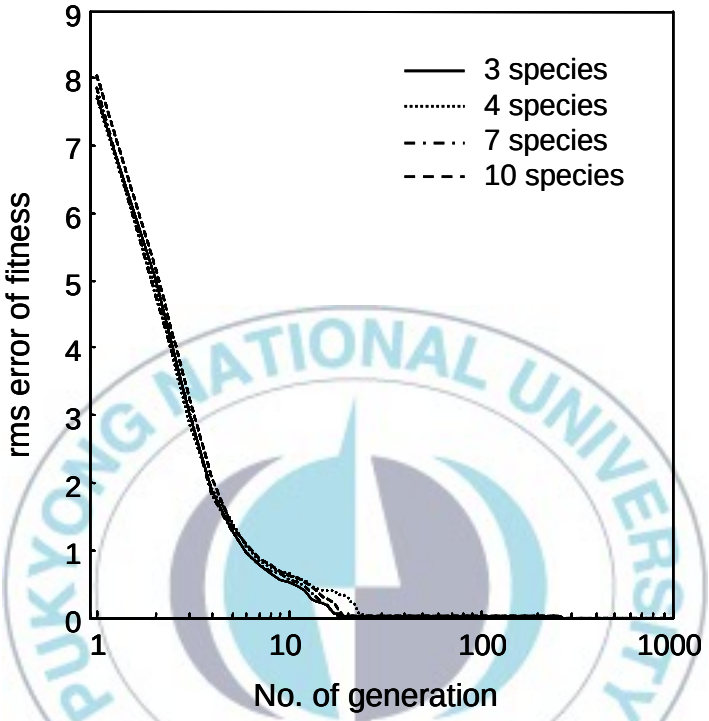
colony at each generation, M is the number of colonies in the artificial world, $f_{j,opt}$ is the optimum values of objective function by exact solution in the jth colony and $f_{j,i}$ is the fitness value of the ith survived individual in the jth colony at each generation. Fig. 2.8 shows the calculating time and the root mean square error of fitness according to generation for various numbers of species. Four global optima are obtained for all the cases considered.

It is estimated that the number of initial individual, N_m, depends on the number of global optimum and now, for the function having 4 global optima, the total global optimum can be found by using just 160 individuals. Although the number of initial individuals is also related with the precision of solution, its effect is minute and it causes an increase of calculation time. So the number of initial individuals defined excessively is inefficient.

A parameter belonging to a category cannot be distinguished strictly by this point of view. One parameter affects one part of behaviors of organisms. But a part of behavior is dominated by some parameters. This research utilizes the classification to determine the parameters for optimum performance in calculating time and accuracy of solutions.



(a) Calculating Time.



(b) rms Error of Fitness.

Fig. 2.8 The Effects of the Number of Species.

6. Numerical Examples and Discussions

The following three test functions, multi-modal function, Eq. (2.16), Rosenbrock banana function, Eq. (2.17), and function with many local optima, Eq. (2.18), are used to verify the effectiveness of the proposed optimization algorithm. These mathematical test functions are often used as a benchmark function for optimization algorithms [13].

6.1. Performance Comparison of EALA and ALA

In this section, calculation results using a multi-modal function are presented for confirming the density of colonies. In general, objective functions with several global and/or local optimum points, are called multi-modal functions. Eq. (2.16) shows a two-dimensional multi-modal function.

$$f(x_1, x_2) = (\cos 2\pi x_1 + \cos 2.5\pi x_1 - 2.1) \times (2.1 - \cos 3\pi x_2 - \cos 3.5\pi x_2) \tag{2.16}$$

This function has four numbers of local and global optima in the given range, respectively. So it is possible to check and estimate the searching performance of the EALA for global optimum and the searching ability of the EALA that can find all global optima simultaneously by this function. The optimum results are **x** = {(0.4388, 0.3058), (0.4388, –0.3058), (–0.4388, 0.3058), (–0.4388, –0.3058)} with f (x) = –16.09172 in the solution space.

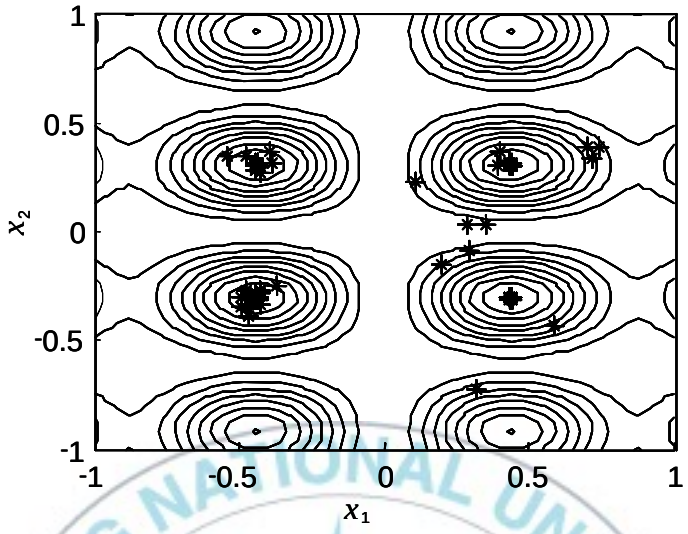
Fig. 2.9 shows the distribution of survived organisms (existing individuals) and the contour line of objective function after 3,000 generations for the ALA and EALA, respectively. The number of existing individuals is about 300 and 220 in each case of the ALA and EALA. It is shown in Fig. 2.9 that both algorithms, the ALA and EALA, can find the global optimum, but the EALA is better than the ALA in the density of colonies being congregated after 3000 generations. It is

possible to redefine the individual density and the density of colonies as the colonized density using the *k*th variation of colony V_k in order to estimate the density of individuals in a colony quantitatively:

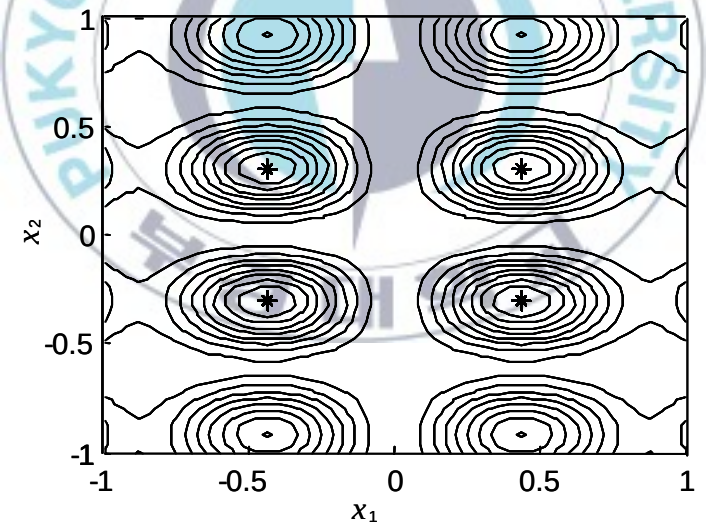
$$V_k = \left[\frac{1}{N} \sum_{j=1}^{M} \sum_{i=2}^{N_j} \left\| x_{k,j,1} - x_{k,j,i} \right\| \right]^{1/2} \tag{2.17}$$

where $N = N_1 + N_2 + \cdots + N_M$, N_j is the number of organisms in the *j*th colony, *M* is the number of colonies in the artificial world (or the domain), $x_{k,j,1}$ is the *k*th coordinate point (or the *k*th design variable) of the best organism in the *j*th colony, $x_{k,j,i}$ is the *k*th coordinate point (or the *k*th design variable) of the *i*th organism in the *j*th colony.

In Fig. 2.9, for the multi-modal function, the 1st variation of the colony is 3.40×10^{-3} and 1.08×10^{-9} for the ALA and EALA, while the 2nd variation of the colony is 2.50×10^{-3} and 1.39×10^{-9} for the ALA and EALA, respectively. Having a small value in the variation of colony means that the distribution of individuals in colonies is dense. As mentioned above, this means a high searching performance.



(a) ALA



(b) EALA

Fig. 2.9 Comparison of Distribution of Survived Organisms at 3000 Generations.

6.2. Performance Comparison of EALA and Other Optimization Methods

$$f\left(x_1, x_2\right) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1^2\right), \left(-2 \le x_1, x_2 \le 2\right) \tag{2.18}$$

$$f\left(x_1, x_2\right) = F\left(x_1\right) \cdot F\left(x_2\right), \left(-2 \le x_1, x_2 \le 2\right) \tag{2.19}$$

where, $F\left(x\right) = \dfrac{\cos(2\pi x) + 1}{2 + 0.2x^2}$, $\left(-2 \le x_1, x_2 \le 2\right)$.

Eq. (2.18) is also called a banana function and has only one global optimum. This function has the phase of hamming cliff. In general, the convergence speed of an evolution program is very slow and the accuracy of a searched solution is low in this phase. Then, it is difficult to find optimum solution because of a valley phenomenon. The optimum solution is given as **x** = (1.0, 1.0) with f (**x**) = 0.

Fig. 2.10(a) shows that the contour line and the emergent colonization are achieved at the global optimum. This function with separable variables has one global optimum and many local optima in the narrow solution space as shown in Eq. (2.19) and Fig. 2.10(b). The global maximum is f (**x**) =1.0 at **x** = (0.0, 0.0).

Four kinds of species of artificial organism are used in searching the optimum solution. Artificial organisms are initially placed at random locations in the solution space, and gradually move towards the global optimum. The artificial organisms that were survived lastly comprise an emergent colonization together with the contour line of solution. The boundaries to which the artificial organism can move during one generation are calculated with Eq. (2.2), where, D_0 = 1.0, α = 12. Each number of the initial artificial organism and the initial resources is 160. The parameters of the updated algorithm are adopted as presented Table 2.1 referring to reference [2]. Fig. 2.10 shows the last surviving artificial organisms in two test functions with a contour line. This figure represents that each individual

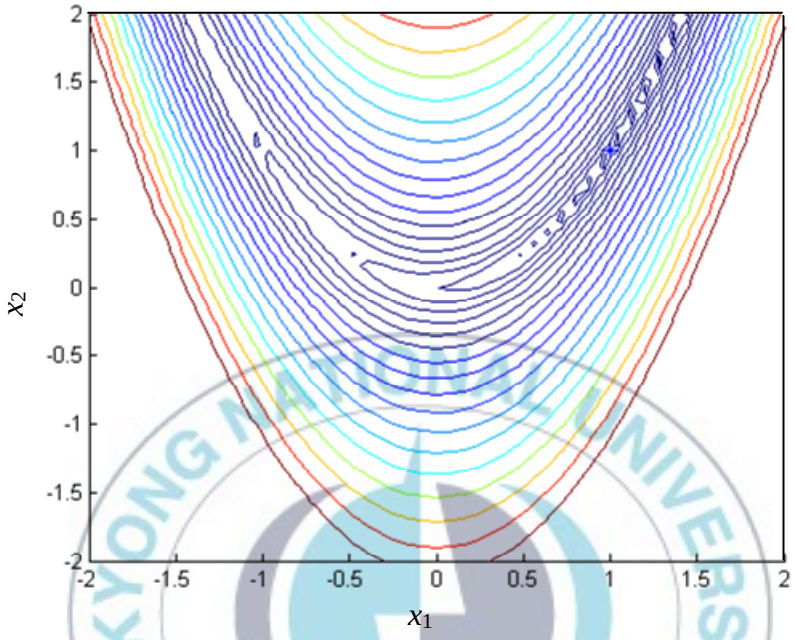
is intensively distributed on optimum solution(s) and is searching all optimum solutions.

The elapsed time and the calculated optimum solutions are shown as Tables 2.2 and 2.3. The optimization results, which are calculated by the EALA, genetic algorithm (GA) [14, 15] and sequential quadratic programming (SQP), are shown in Table 2.2. The SQP is a numerical technique wherein an approximate solution is sought by proceeding in an iterative manner by starting from an initial solution. The GA and EALA can search all global optimum solutions. However, the EALA can get a more fast than GA. Table 2.3 shows the comparison with conventional algorithm (ALA) and EAL for Rosenbrock function and function with many local optimums. The EALA does not only converge faster than the LA, but also can find a more accurate solution. Also, the calculation errors are estimated by Eq. (2.20), and error E according to generations is described in Fig. 2.11.

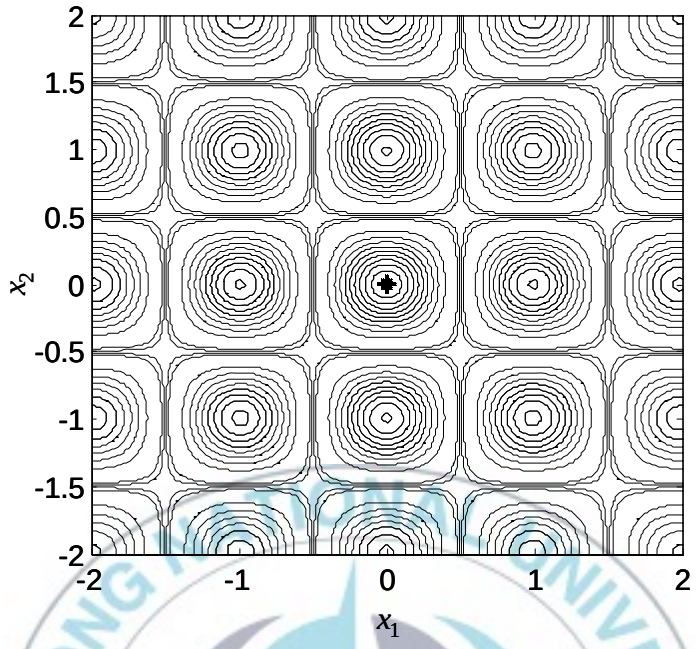
$$E = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{P}\left|\frac{(x_{opt,i} - x_j)}{x_{opt,i}}\right| \tag{2.20}$$

where N is the total number of artificial organisms, and P is the number of the global optimum solutions, and $\mathbf{x}_{opt,i}$ is ith optimal variable.

Fig. 2.11 shows each the convergence characteristic of ALA and EALA. It is verified that accuracy of the solution by the EALA is superior to the existing artificial algorithm. Also, in searching for the optimum solution of Rosenbrock function, the ALA needs 49 seconds, but the EALA needs only 1 second (Table 2.2). In other example functions, the new algorithm also has very good performance in the aspects of time which the algorithm needs for reaching the precision which the before algorithm has after 3000 generations (Table 2.3). The solution accuracy after 3000 generations, which the EALA has, is better than that of the before existing algorithm.

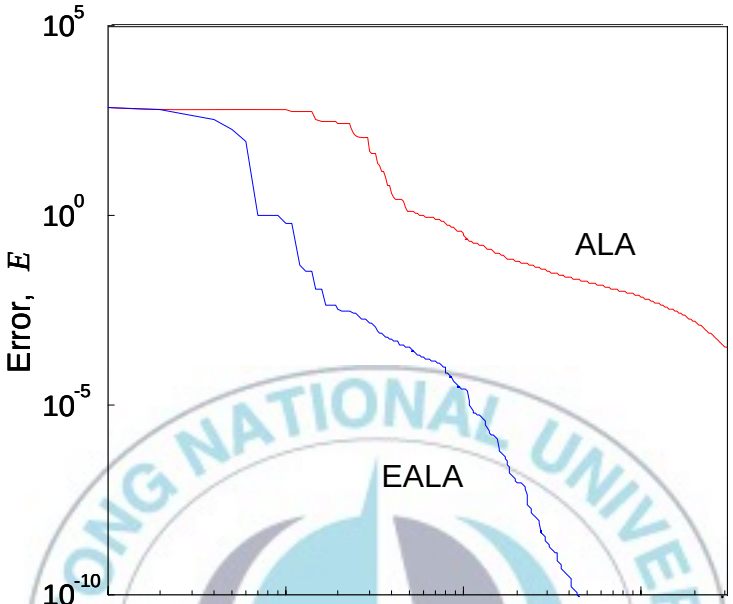


(a) Function Defined by Eq. (2.18)

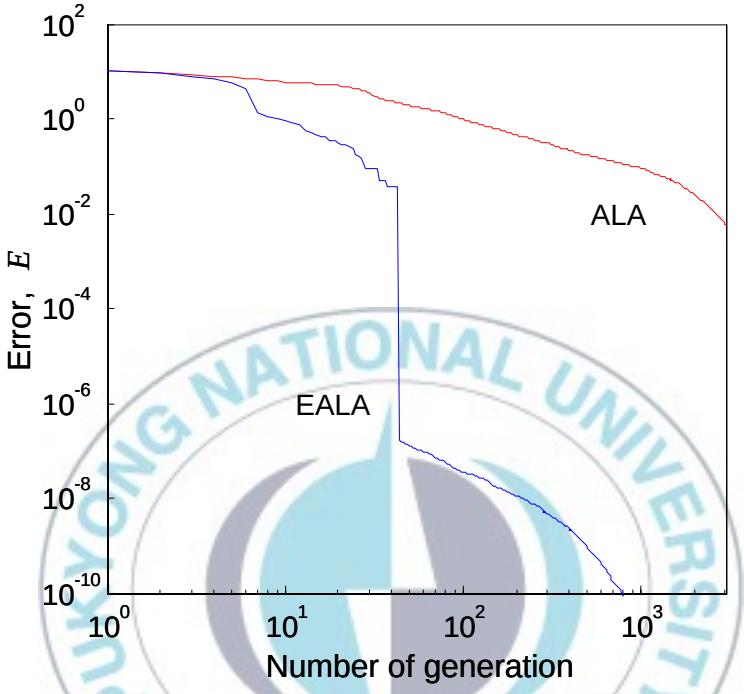


(b) Function Defined by Eq. (2.19)

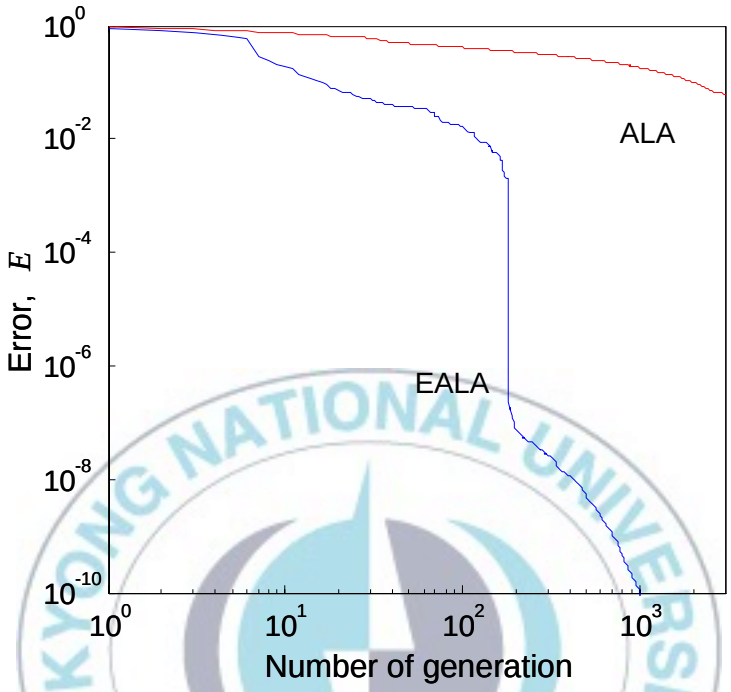
Fig. 2.1 Contour Line and Emergent Colonization after Final Generation.



(a) Rosenbrock Function



(b) Multi-modal Function



(c) Function with Many Local Optimums

Fig. 2.11 Convergence Characteristics of Colonization to Optimum Solution.

Table 2.2 Comparison of Optimization Results for Multi-modal Function

Methods	Optimum value	Number of generation	Computing time (s)
ALA	-16.09172	3000	17.7
EALA	-16.09171	28	0.26
GA	-16.09172	26	8.41
SQP ($\boldsymbol{x}_s$ = 0.5, 1)	-14.33086	-	0.48

Table 2.3 Comparison of Optimization Results for Rosenbrock Function and Function with Many Local Optimums

Methods	Optimum value		Number of generation		Computing time (s)	
	ALA	EALA	ALA	EALA	ALA	EALA
Rosenbrock	0.0	0.0	3000	101	17.9	0.75
Local optima function	0.9999	1.0	3000	251	15.6	2.18

7. Conclusions

The parameters of the EALA have been simplified. The effectiveness of the EALA is evaluated using three test functions. The optimized results are compared with those of conventional artificial life algorithm. The results show that the EALA reaches the optimum solution faster and closer to exact solution than the conventional artificial life algorithm, and will give all global optimum solutions in a solution space. The algorithm reduced the time needed for searching, to say nothing of accuracy of optimum solutions, because the ELAL improved the method of making decision of new points for offspring and for own waste by using random tabu search method.

References

[1] T. Satoh, A. Uchibori, K. Tanaka, 1999, “Artificial Life System for Optimization of Nonconvex Functions,” Proceedings of International Conference on Neural Networks, 4, 2390-2393.

[2] B.S. Yang, Y.H. Lee, 2000, “Artificial Life Algorithm for Function Optimization,” ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference. Baltimore, DETC 2000/DAC-14524.

[3] B.S. Yang, Y.H. Lee, B.K. Choi, H.J. Kim, 2001, “Optimum Design of Short Journal Bearings by Artificial Life Algorithm,” Tribology International 34(7), 427-435.

[4] C. Yang, H. Ye, J.C. Wang, L. Wang, 2002, “An Artificial Life and Genetic Algorithm Based on Optimization Approach with New Selecting Methods,” Proceedings of International Conference on Machine Learning and Cybernetics, IEEE, Beijing, 2, 684-688.

[5] S.S. Rao, 1996, Engineering Optimization, Theory and Practice, John Wiley & Sons, Inc.

[6] F. Glover, 1989, “Tabu Search- Part I,” ORSA Journal on Computing, 1(3), 190-206.

[7] F. Glover, 1990, “Tabu Search- Part II,” ORSA Journal on Computing, 2(1), 4-32.

[8] H. Youssef, S.M. Sait, H. Adiche, 2001, “Evolutionary Algorithms, Simulated Annealing and Tabu Search: a Comparative Study,” Engineering Applications of Artificial Intelligence, 14, 167-181.

[9] J. Hu, H. Hu, Y. Fang, 1996, “Research into Models and Algorithms of Artificial Life,” Artificial Intelligence in Engineering, 10, 299-310.

[10] Y.K. Ahn, J.D. Song, B.S. Yang, 2003, “Optimal Design of Engine Mount

Using an Artificial Life Algorithm," Journal of Sound and Vibration, 261, 309-328.

[11] B.S. Yang, J.D. Song, 2001, "Enhanced Artificial Life Algorithm for Fast and Accurate Optimization Search." Proceedings of A-PVC, 732-736.

[12] B.S. Yang, B.G. Choi, Y.H. Yu, H.T. Nan, 1999, "Optimum Design of a Damping Plate with an Unconstrained Viscoelastic Damping Layer Using Combined Genetic Algorithm," KSME International Journal, 13(5), 387-396.

[13] A. Homaifar, C. Qi, S. Lai, 1994, "Constrained Optimization via Genetic Algorithm Simulation," Electronics Letter, 62(4), 242-254.

[14] B.G. Choi, B.S. Yang, 2000, "Optimum Shape Design of Shaft Using Genetic Algorithm," Journal of Vibration and Control, 6(2), 207-222.

[15] D.E. Goldberg, 1989, Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley Publishing Company.


PUKYONG NATIONAL UNIVERSITY
부경대학교

Ⅲ. Pareto Optimization Algorithm Artificial Life (PAL)

1. Introduction

Most engineering optimization problems often consist of several objective functions rather than a single objective function. There are basically two kinds of approaches to solve a multi-objective optimization method. First approach transforms a given multi-objective optimization problem (MOP) into a single objective optimization problem (SOP). One method of this approach is to aggregate multiple objective functions into a single overall objective function. Optimization of the objective function is then conducted with one optimal design as a result. This result is greatly dependent on how the objectives are aggregated [1]. One of the two forms such as linear combination or multiplication is usually employed as an aggregated single objective function. Another method is to select only the most interesting object function as a final objective function and to set the other objective functions as constraints.

The motivation of the first approach is basically to establish a single basis of comparing each candidate solution in the course of optimization and finally to derive a single optimum solution (or approximate to the optimum solution).

The second approach is simultaneously to consider the multiple objective functions, which is called Pareto optimization.

In order to provide possible solutions for the final decision maker, this approach is to supplement the downside which is not able to find other possibilities besides a single solution obtained through conversion into a single objective function. In order to avoid this difficulty and to explore various

possibilities, the concept of Pareto optimality is employed.

Many researches on Pareto optimization problems have been carried out recently to enable the application of heuristic global optimization algorithms such as evolutionary algorithm [2-6] and tabu search method [7, 8]. In the case of function optimization, heuristic optimization methods have the advantages of not being subjected to special restrictions on problem formulations. They are also evaluated as having outstanding search capability in finding a global optimum solution of optimization. As a heuristic global optimization technique, artificial life algorithm [9-11] has been applied to determine optimum design problems of journal bearing [12] and engine mount [13, 14]. However, the expansion onto Pareto optimization has not yet been attempted in real applications.

In order to apply artificial life algorithm to MOP in engineering problems, it is necessary to solve the Pareto optimization problem. Therefore, in this chapter, artificial life optimization algorithm has been expanded to enable the application of Pareto optimization to solve the MOPs.

2. Multi-objective optimization problems (MOP)

A MOP is defined as a problem which has two or more objective functions. A general MOP is defined as

$$\text{minimize} \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T \tag{3.1}$$

$$\text{subject to} \quad \mathbf{c}(\mathbf{x}) = (c_1(\mathbf{x}), c_2(\mathbf{x}), \ldots, c_m(\mathbf{x}))^T \geq 0 \tag{3.2}$$

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T, \quad \mathbf{x} \in S \tag{3.3}$$

where $f_i(\mathbf{x})$ is the set of k objective functions, $c_i(\mathbf{x})$ is the set of m constraints, x_j is the n optimization parameters, and $S \in R^n$ is the solution or parameter space. Obtainable objective vectors $\{\mathbf{F}(\mathbf{x}) \mid x \in S\}$ are denoted by Y, where $Y \in R^k$ is usually referred to as the attribute space.

In MOP, it is important to emphasize that there might be constraints imposed on the objectives. It is normal for the objectives of MOP to be in conflict with each other [15]. However, most MOPs do not lend themselves to a single solution but have a set of solutions. Such solutions are trade-offs or good compromises among the objectives. In order to generate these trade-off solutions, an old notion of optimality is normally adopted. This notion of optimality was generalized by Pareto [16] and is called Pareto optimum. The solution to a MOP is Pareto optimal if there is no other feasible solutions which would decrease some objective function values without causing a simultaneous increase in at least one other objective function value.

3. Pareto Optimization

Let's consider a minimization problem which has two or more objective functions. A Change in design variables (or design vector) in order to lower the value of an objective function may generally result in the increased values of other objective functions. Therefore, in most cases, a set of solutions that simultaneously minimize all the objective functions becomes a null set. This problem leads to a new concept called Pareto set.

The Pareto set consists of solutions that are not dominated by any other solutions. Considering a minimization problem with two solution vectors **x** and **y** $\in S$. Where **x** is said to dominate **y**, and is denoted by $\mathbf{x} \prec \mathbf{y}$, if:

$$\forall i \in \{1, 2, \ldots, k\} : f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \text{ and } \exists j \in \{1, 2, \ldots, k\} : f_j(\mathbf{x}) < f_j(\mathbf{y}) \qquad (3.4)$$

The space in R^k formed by the objective vectors of Pareto optimal solutions is known as the Pareto optimal front.

Let's consider a minimization problem to hold two objective functions, f_1 and f_2. For the two design vectors of A and B, the case in which all objective function values of A are the same or smaller than all objective function values of B and also at least one objective function value is smaller than B is described as "B is dominated by A". In the case in which the first objective function value of A, $f_1(A)$, is smaller than $f_1(B)$, but the second objective function value, $f_2(A)$, is larger than $f_2(B)$, where A and B are referred to as the "non-dominated solution".

Ultimately, Pareto optimization problem is considered to find the set of all non-dominated solutions and this set of solutions is referred to as Pareto set of solutions. Also, the set of objective function values in the range by Pareto set of solutions are referred to as Pareto front. A range holds the same dimension as the number of objective functions. If there are four or more objective functions, it is difficult to express them geometrically. Therefore, this study only deals with

simple problems with two design variables and two objective functions in order to validate and confirm the proposed method.

4. Pareto Artificial Life Optimization Algorithm (PAL)

For convenience, artificial life optimization algorithm for a single objective function [8] is called artificial life algorithm (ALA), and the artificial life optimization algorithm used to find the optimum Pareto solution is called Pareto optimization algorithm based on artificial life (PAL). For ALA, the course of finding solution imitates cluster formation, which is one of the ecological processes of natural phenomena. The basic concept is to promote cluster formation in proximity in order to optimize the solution and carry out an extensive search in the cluster. For PAL, the fitness evaluation method in ALA has been improved to suit multi-objective function. Also, by adding the Pareto list, adjustment has been made to enable application to multi-objective function optimization problem.

4.1. Fitness Evaluation

In ALA, cluster formation by artificial objects has been promoted in areas with outstanding fitness within the space of design variables by evaluating the fitness based on objective function values. In the case of multi-objective function optimization problem, changes in design variables generally result in a reduction of one objective function value and an increase in another objective function value. Therefore, a new method must be found for fitness evaluation.

Horn et al. [5] attempted to solve Pareto optimal design problem with evolutionary algorithm by proposing a modified shared fitness evaluation method which introduced the concept of niche into the concept of shared fitness by Goldberg [4].

Shared fitness by Horn et al. is found by dividing the ordinary fitness used in genetic algorithms (GA), with the number of niches. Niche is defined as biological position in the study of biology. The detailed concept of niche used in

GA can be interpreted as a type of density of solutions. Ho et al. [8] modified the shared fitness in order to apply it in tabu search method to Pareto optimization problem. The shared fitness by Ho et al. uses the concept of the density of solutions. In Ho et al. algorithm, the difference is that they did not use the ordinary fitness rules used in GA in order to use the tabu search method, instead the reciprocal number of the density of solutions is considered. Also, when finding the number of niches, Horn et al. [5] evaluated and reflected the Euclid distance d_{ij} of two objects, i and j defined in the space of objective functions. However, they evaluated the density with only the number of solutions within the set proximity.

In PAL, shared fitness is improved by using density evaluation method reflected with distances among solutions in the concept of shared fitness as proposed by Ho et al. [8]. In detail, shared fitness of PAL is defined as of Eq. (3.5).

$$f_{share(i)} = \frac{1/d_i}{\sum_{j=1}^{NAC_i} 1/d_j} \tag{3.5}$$

$$d_j = \frac{NAC_j}{RC_j} \tag{3.6}$$

$$RC_j = \begin{cases} = \dfrac{\sum_{k=1}^{NAC_j-1} \left\| \mathbf{F}^{(k)} - \mathbf{F}^{(j)} \right\|}{NAC_j - 1} & \text{if } NAC_j = 1 \\ = RC_0 & \text{else} \end{cases} \tag{3.7}$$

where, f_{share}, d_j, RC_0, RC_j and NAC_j represent shared fitness, density of solutions of object j (or candidate solution), radius to calculate density d_j on the basis of the

location of object j, radius of the final proximity used in calculating d_j and number of objects including itself in proximity to the object j defined with RC_0 respectively.

In shared fitness proposed by Ho et al., RC_j is a fixed value regardless of the distribution status of solutions or objects and corresponds with RC_0 in this study. Therefore, in evaluating the density of objects around a specific object, depending only on the number of solutions within a domain regardless of the status of distribution within a specific domain, detail approach was presented by Ho et al. [8]. However, for PAL, by introducing RC_j, density evaluation of solutions could be further improved not only by the number of solutions in the proximity set by user, but also by evaluating the distance between these solutions and the center point.

4.2. Pareto Set of Solutions

The concept of Pareto can be interpreted as a type of long-term memory collection of past memory set in PAL such as tabu search method. To find Pareto set of solutions, Pareto archive is renewed by adding and replacing non-dominated solutions among all artificial objects of each generation. It is also renewed in each generation.

4.3. Flowchart of PAL

The flow diagram of the algorithm to describe the process of the proposed PAL is shown in Fig. 3.1. The overall processes are explained as follows.

Step 1: Initial artificial life objects and resources are distributed randomly in solution space and granted initial internal energy for each organism.

Step 2: All artificial objects conduct search, metabolism, movement and reproduction.

Step 3: Artificial life object searches the resources to exist within proximity

domain of its own to hold the radius as defined by Eq. (3.8),

$$D = e^{-\alpha(G/G_{\max})} \tag{3.8}$$

where, G, $G_{\max}$ and α, respectively, represent the number of current generations, the maximum number of generations and factor to reduce the radius of proximity domain according to generation. In particular, α was reviewed by Yang and Lee [9] and it was set as $\alpha = 3$ in this study.

Metabolism: When resources are searched, artificial life object moves to the most closely located resource and metabolizes the closest resource. The metabolized resource is eliminated and, object increases internal energy. It also disposes waste in a random position within the proximity domain.

Movement: In the case where resources are not found in proximity, a random location is selected within the proximity. If the selected location has a higher fitness than the current location, the artificial life object moves to the selected location. If not, the process of randomly selecting new locations is repeated.

Reproduction: For objects of which the internal energy are over the minimum energy to enable reproduction, the closest object among artificial life objects of the same species is selected as itself within the proximity domain. If the same conditions are satisfied, the object carries out reproduction. By reproduction, two new objects are created as offspring. Initial location of each of these two objects is decided as random location with higher fitness

than the fitness of each of their parent objects within proximity domain for locations of each of the parent objects.

Step 4: Renew Pareto archives with the method described in section 4.2.

Step 5: Increase the number of generation and age of organism by 1.

Step 6: When internal energy is of critical value as internal energy reduced, artificial life object is considered extinct and is deleted.

Step 7: Return to step 2 in case where the number of generations does not reach the maximum value, and if it reaches the maximum value, end the process.

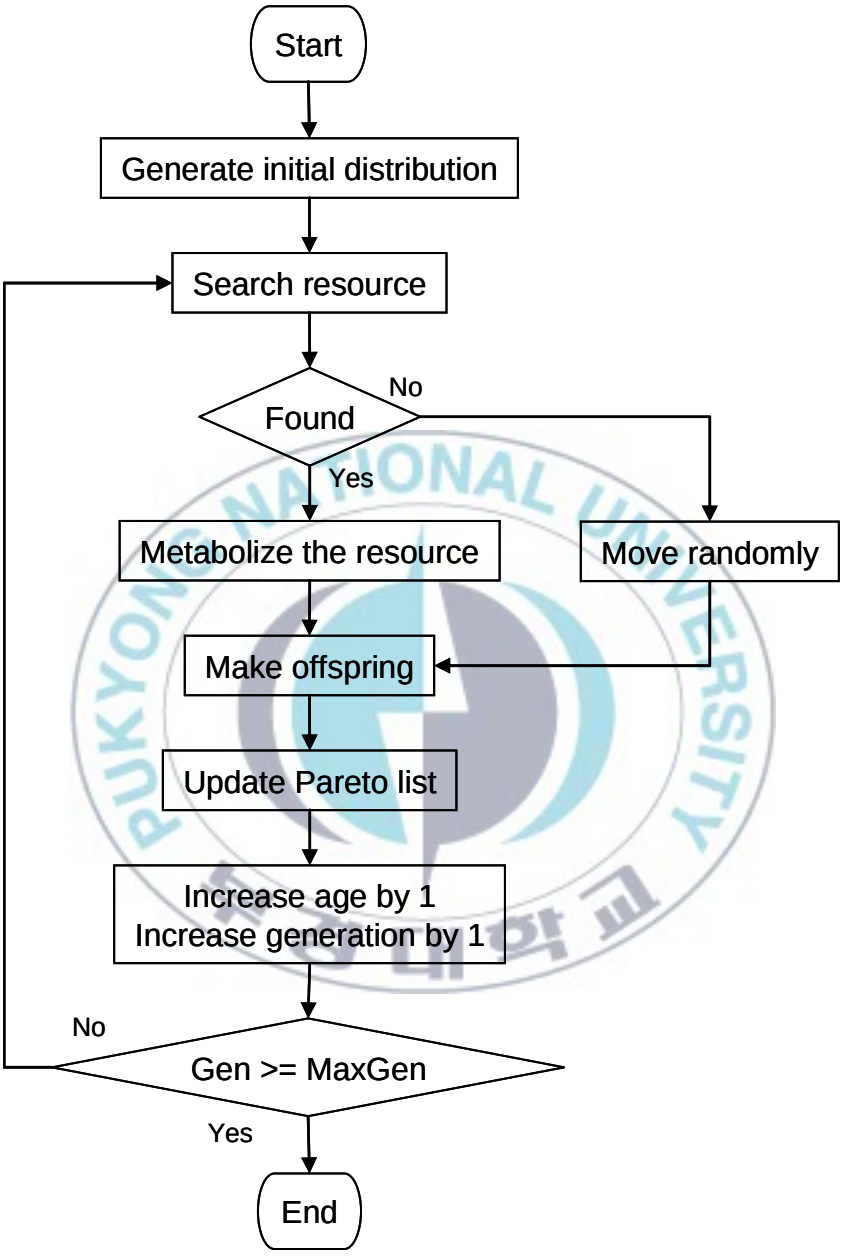


Fig. 3.1 Flowchart of Pareto Artificial Life Optimization Algorithm.

5. Examples

The performance verification is carried out by applying PAL to three test functions, of which the optimum solutions were known. The first test function [17] is defined by Eq. (3.9) and, for convenience, n, the number of design variables, was set as 2.

Minimize

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T \tag{3.9}$$

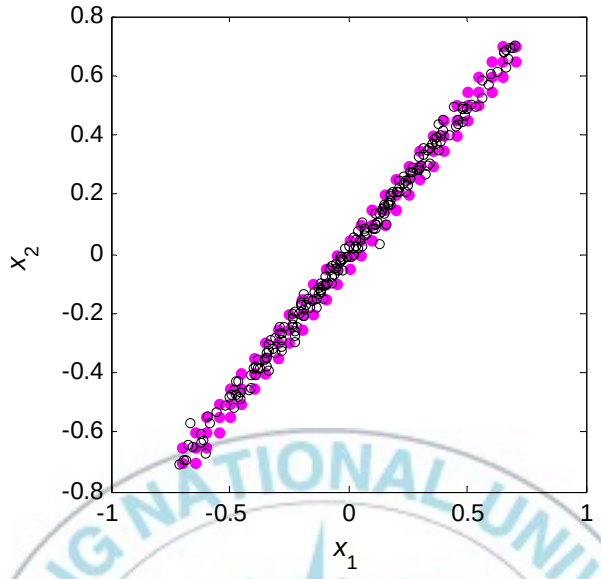
$$f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right),$$

$$f_2(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right)$$

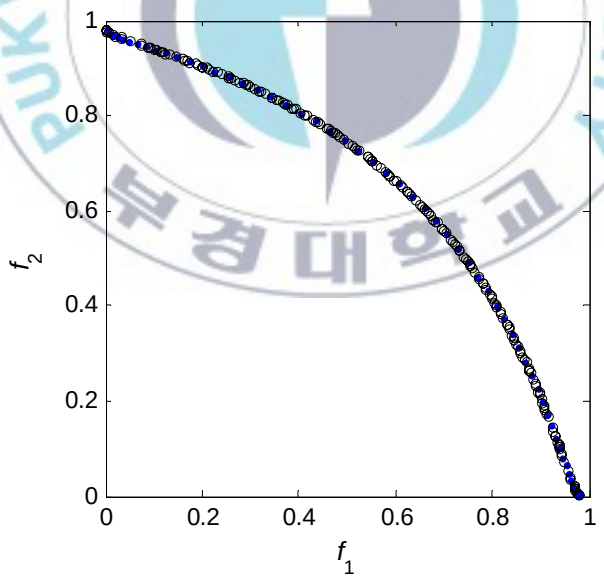
Subjected to:

$$-2 \le x_i \le 2$$

There are no constraints with the exception of the upper and lower limits of design variables. Pareto set of solutions and Pareto front are shown in Fig. 3.2 together with the results by PAL. The "dots" represent Pareto set of solutions and "○" marks are the solutions obtained by PAL.



(a) Pareto Set



(b) Pareto Front

Fig. 3.2 Pareto Set and Pareto Front of Test Function 1.

The second test function [6] is defined by Eq. (3.10).

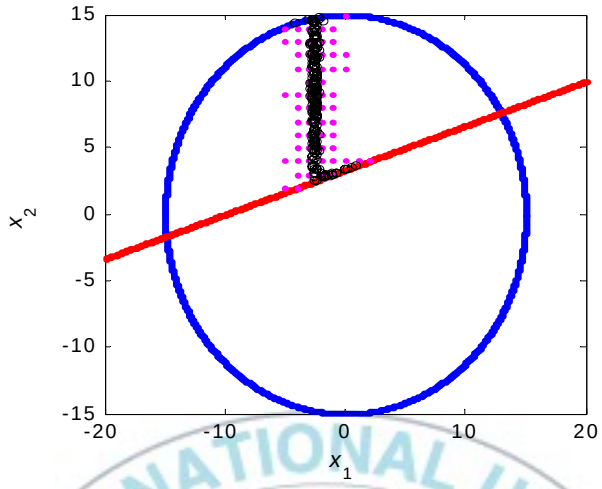
Minimize

$$\begin{aligned}
&\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T \\
&f_1(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \\
&f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2
\end{aligned} \tag{3.10}$$

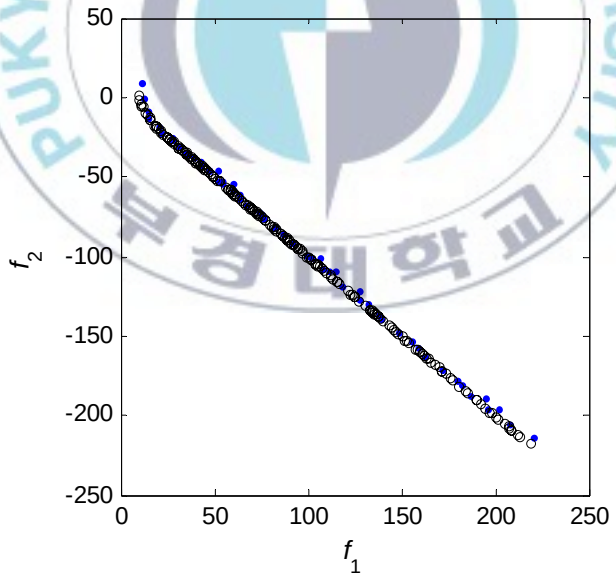
Subjected to

$$\begin{aligned}
&c_1(\mathbf{x}) \equiv x_1^2 + x_2^2 - 225 \le 0, \quad c_2(\mathbf{x}) \equiv x_1 - 3x_2 + 10 \le 0 \\
&-20 \le x_1, x_2 \le 20
\end{aligned}$$

There are two constraints in addition to the upper and lower limits of design variables. Feasible solutions are the intersections at the top of straight line and inside the circle shown in Fig. 3.3(a). In Fig. 3.3(a), Pareto set of solutions is expressed with dots. The overlapping "○" marks represent Pareto set of solutions calculated by PAL. Pareto front is shown in Fig. 3.3(b). From the second example, it is confirmed that Pareto set of solutions could be successfully found while satisfying the constraints in cases where there were simple constraints.



(a) Pareto Set



(b) Pareto Front

Fig. 3.3 Pareto Set and Pareto Front of Test Function 2.

The third test function [18] is defined by Eq. (3.11).

Minimize

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T$$
$$f_1(\mathbf{x}) = x_1,\ f_2(\mathbf{x}) = x_2 \tag{3.11}$$

Subjected to

$$c_1(\mathbf{x}) \equiv x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\tan^{-1}\frac{x}{y}\right) \le 0$$
$$c_2(\mathbf{x}) \equiv (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \le 0$$
$$0 \le x_1, x_2 \le \pi$$

The small insert at the bottom left-hand corner of Fig. 3.4 shows the Pareto solutions. According to the two constraints, feasible solutions are those which correspond to the part filled with slashes and include the boundary. Here, Pareto solutions of Fig. 3.4 are shown by thick lines in the small insert and are based on the first constraint c_1. Among the feasible solutions marked with slashes, those shown with dotted lines are active constraints and therefore are feasible solutions. However, they cannot be Pareto set of solutions as they are not the non-dominated solutions. In particular, this problem is useful in evaluating the search capabilities in relations to the problems in which active constraints exist. It is also confirmed that PAL has successfully found the set of solutions.

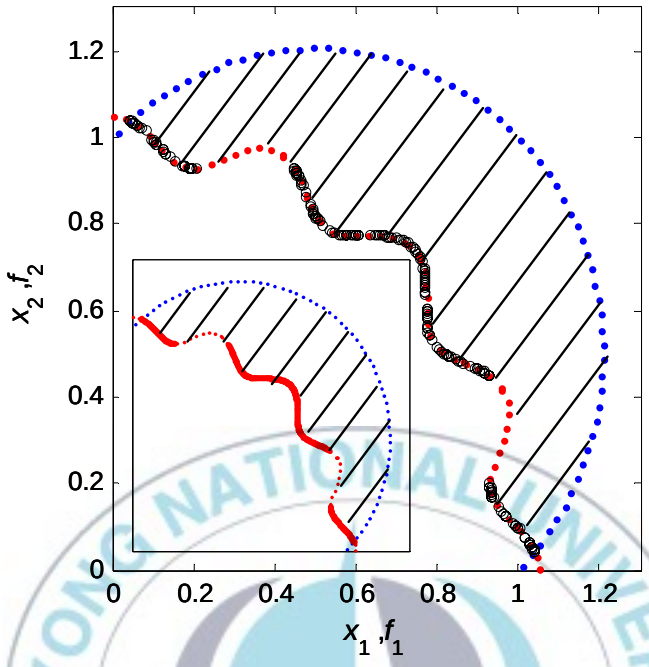


Fig. 3.4 Pareto Set and Pareto Front of Test Function 3.

6. Pareto optimum design of journal bearing

Pareto optimization problem of high-speed and short journal bearing shown in Fig. 3.5 is considered with the two objective functions, temperature increase and supply flow.

Optimum design of journal bearing has been studied by many researchers. Hashimoto [19], Yang et al. [10] and Song et al. [12] studied the optimization by using sequential quadratic programming (SQP), the ALA and the EALA, respectively. However, they only searched the optimization of a single objective function and only as the form of linear combination of two objective functions, temperature increase and supply flow. Therefore, this study intends to provide Pareto set of solutions by carrying out Pareto optimization.

6.1. Defining state variables of journal bearing

As state variables used in the objective functions or constraints, bearing load W (N), operating speed n_s (rps), eccentricity ratio ε_0, oil film pressure p (MPa), oil film temperature T (°K), journal surface friction F_j (N), supply flow Q (m^3/s) and whirling onset speed ω_{cr} (rad/s) are considered.

These state variables, in general, are determined by design variables. For design variables, the radial clearance C and bearing width to diameter ratio λ (= L/D) are considered. The state variables, equations (3.12) ~ (3.19), are obtained from Hashimoto [19].

The average Reynolds number is defined as of $R_e(\mathbf{x}) = \rho CU / \mu$, where μ represents the viscosity. The correction coefficients α_m and G_θ^* are defined by the following equations according to Reynolds number.

$$R_e < 510: \alpha_m = 1, \quad G_\theta^* = 1/12$$

$$510 \le R_e < 1125: \quad \alpha_m = 5.914 R_e^{-0.285}, \quad G_\theta^* = 2.915 R_e^{-0.57}$$

$$1125 \le R_e < 13500: \quad \alpha_m = 0.798, \quad G_\theta^* = 2.915 R_e^{-0.57}$$

$$R_e \ge 13500: \quad \alpha_m = 0.756, \quad G_\theta^* = 14.45 R_e^{-0.75} \tag{3.12}$$

Modified Sommerfeld number S is the most important factor in bearing design, and eccentricity ratio ε_0 expressed as the function of S, is obtained from the following equation.

$$S = \frac{n_s \mu D^3 \lambda}{48 G_\theta^* C^2 W}, \text{ and } \varepsilon_0 = \exp(-2.236 \alpha_m \lambda \sqrt{S}) \tag{3.13}$$

The minimum oil film thickness $h_{\min}$, whirling start speed ω_{cr} and the maximum oil film pressure $p_{\max}$ under the steady-state condition are respectively obtained from Eqs. (3.14), (3.15) and (3.16).

$$h_{\min} = C(1 - \varepsilon_0) \tag{3.14}$$

$$\omega_{cr} = \left[0.0584 \exp\left(6.99 \varepsilon_0^{2.07}\right) - 1.318 \varepsilon_0 + 2.87 \right] (g/C)^{1/2} \tag{3.15}$$

$$p_{\max} = \frac{\pi n_s \mu D^2 \alpha_m^2 \lambda^2}{8 G_\theta^* C^2} \frac{\varepsilon_0 \sin\theta_0}{(1 + \varepsilon_0 \cos\theta_0)^3} \tag{3.16}$$

where g represents acceleration of gravity, θ_0 represents the angular location of the maximum oil film pressure occurs and is defined by Eq. (3.17).

$$\theta_0 = \cos^{-1}\left(\frac{1 - \sqrt{1 + 24\varepsilon_0^2}}{4\varepsilon_0} \right) \tag{3.17}$$

The journal surface friction is approximately given by Eq. (3.18) according to Reynolds number.

$$R_e<1125:\quad F_j \cong \frac{\pi^2 \mu n_s D^3 \lambda}{48 G_\theta^* C}\left\{\frac{1}{\sqrt{1-\varepsilon_0}}+\frac{1-\varepsilon_0}{(1-\varepsilon_0^2)^{3/2}}\right\}$$

$$1125 \le R_e<13500:\quad F_j \cong \frac{\pi^2 \mu n_s D^3 \lambda}{48 G_\theta^* C}(1.109\varepsilon_0^2-1.490\varepsilon_0+2.748)$$

$$R_e \ge 13500:\quad F_j \cong \frac{\pi^2 \mu n_s D^3 \lambda}{48 G_\theta^* C}\left(1.792\varepsilon_0^3-1.523\varepsilon_0^2-3.697\varepsilon_0+8.734\right) \tag{3.18}$$

Lastly, flow of supplied lubricant Q and oil film temperature increase ΔT can be found from Eq. (3.19).

$$Q=\frac{\pi}{4} n_s C D^2 \varepsilon_0 \quad , \quad \Delta T=\frac{F_j D \omega}{2\rho C_p Q}=\frac{2F_j}{\rho C_p D C \varepsilon_0} \tag{3.19}$$

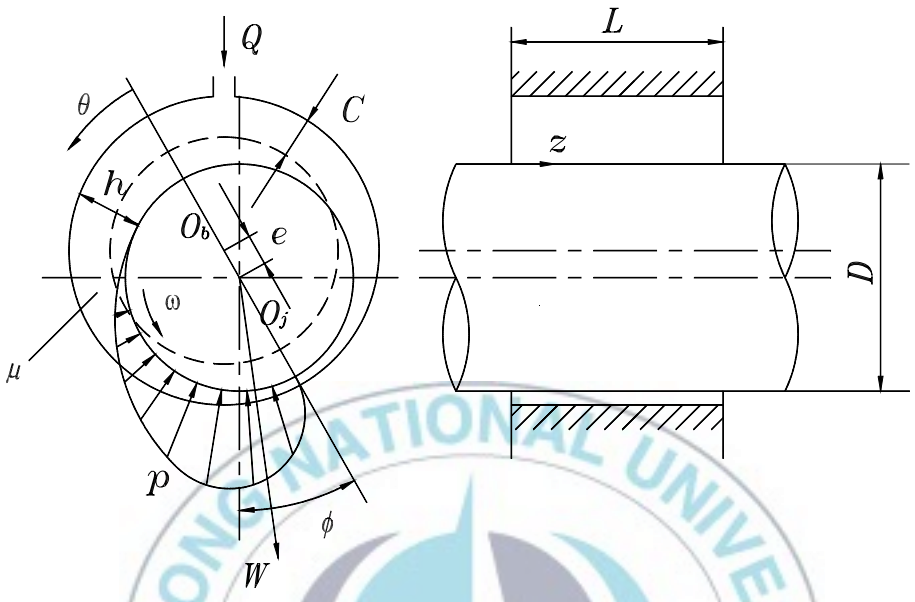


Fig. 3.5 Geometry of a Hydrodynamic Journal Bearing.

Table 3.1 Input Parameters for Optimum Design

Minimum radial clearance	C_{min} = 40 μm
Maximum radial clearance	C_{max} = 300 μm
Minimum length to diameter ratio	λ_{min} = 0.2
Maximum length to diameter ratio	λ_{max} = 0.6
Lubricant viscosity	μ = 0.001 Pa·s
Allowable minimum film thickness	h_a = 10 μm
Allowable maximum film pressure	p_a = 10 MPa
Allowable film temperature rise	ΔT_a = 70 °K
Density of lubricant	ρ = 860 kg/m^3
Specific heat of lubricant	C_p = 4.19×103 J/kg·°K

6.2. Optimum design formulation

Optimization problem is formulated by the basis of state variables described in section 6.1. Design variables are the radial clearance C and width to diameter ratio λ. Design variable vector is defined by Eq. (3.20). The objective function vector for the supply flow and temperature increase is defined by Eq. (3.21). Also, the constraints for the design variables are defined by Eq. (3.22).

$$\mathbf{x}=(C,\lambda)^T \tag{3.20}$$

$$\mathbf{F}(\mathbf{x})=(Q(\mathbf{x}),\ \Delta T(\mathbf{x}))^T \tag{3.21}$$

$$c_i \equiv g_i(\mathbf{x})\le 0,\quad (i=1\sim 8) \tag{3.22}$$

$$\begin{aligned}
&g_1 = C_{\min} - C,\quad g_2 = C - C_{\max},\\
&g_3 = \lambda_{\min} - \lambda,\quad g_4 = \lambda - \lambda_{\max},\\
&g_5 = h_a - C\{1-\varepsilon_0(\mathbf{x})\} = h_a - h_{\min},\\
&g_6 = \Delta T(\mathbf{x}) - \Delta T_a,\quad g_7 = \omega - \omega_{cr}(\mathbf{x}),\\
&g_8 = p_{\max}(\mathbf{x}) - p_a
\end{aligned}$$

In g_1 to g_4 of Eq. (3.22), the subscripts of min and max, respectively, represent the lower and upper limit values of the design variables. In order to observe the changes in Pareto front and Pareto set of solutions according to major state variables, optimization is carried out by changing the operating speed n_s and load W exerted on bearing.

Lastly, to investigate the relevance between Pareto set of solutions and optimum solutions obtained from changing this problem into a single objective function in the form of linear combination, optimization problem with one objective function of Eq. (3.23) is

$$F(\mathbf{x}) = \alpha_1\beta_1 Q(\mathbf{x}) + \alpha_2\beta_2\Delta T(\mathbf{x}) \quad \text{, and } \left(\alpha_1+\alpha_2=1\right) \tag{3.23}$$

where, α_1 and α_2 are weighting factors.

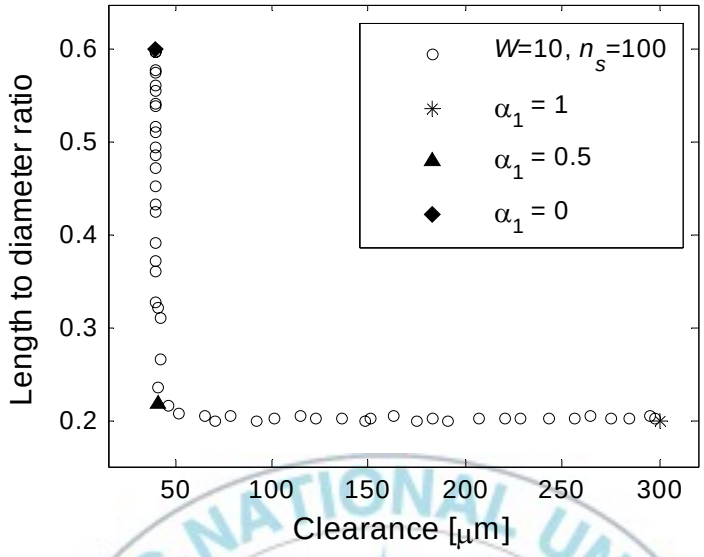
In particular, for comparison with Pareto set of solutions, scale factors are fixed as $\beta_1 = 1,\ \beta_2 = 1/5000$ and the weighting factors are considered as the following:

$$(\alpha_1, \alpha_2) = \{(1.0, 0.0), (0.8, 0.2), (0.6, 0.4), (0.5, 0.5), (0.4, 0.6), (0.2, 0.8), (0.0, 1.0)\} \tag{3.24}$$

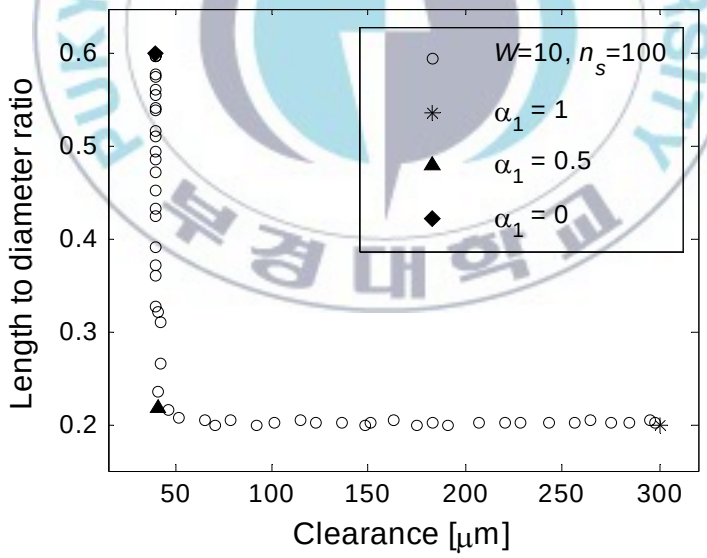
Then, optimization is carried out for each case.

6.3. Results of optimum design

Result of optimization by the PAL is presented by the symbol '○' and the result of optimization through Eq. (3.23) by the EALA is presented by the symbols of '*', '▲' and '◆', etc. The results are shown in Figs. 3.6 and 3.7.

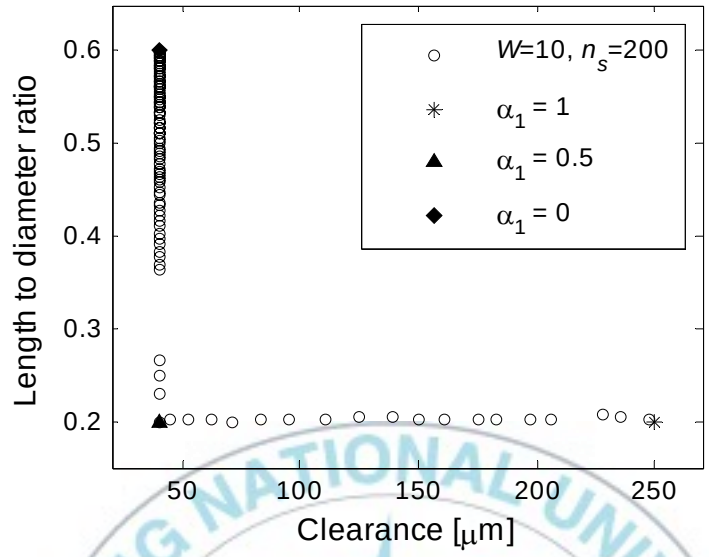


(a) Pareto Set

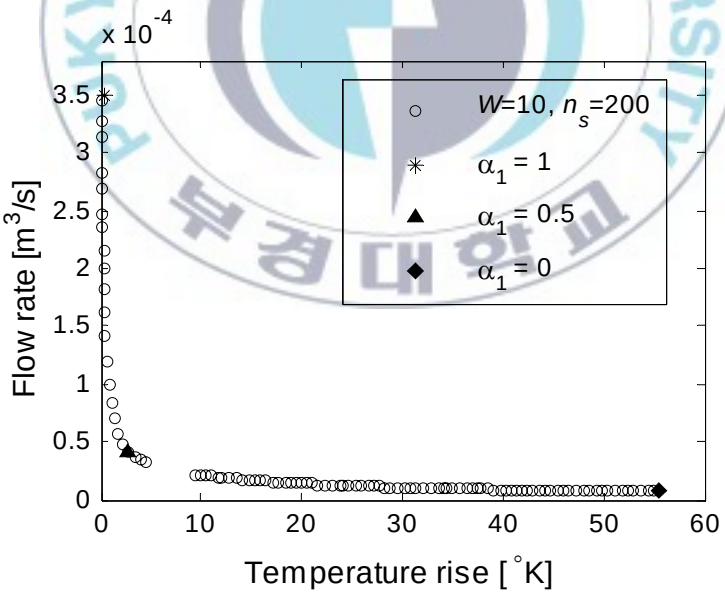


(b) Pareto Front

Fig. 3.6 Comparison of Pareto Optimal and Single Optimal $((\alpha_1,\alpha_2)=\{(1.0,0.0),(0.5,0.5)(0.0,1.0)\}$, W = 10 kN, n_s = 100 rps).



(a) Pareto Set



(b) Pareto Front

Fig. 3.7 Comparison of Pareto Optimal and Single Optimal $((\alpha_1,\alpha_2)=\{(1.0,0.0),(0.5,0.5)(0.0,1.0)\}$, W = 10 kN, n_s = 200 rps).

Fig. 3.6(a) and Fig. 3.7(a) express the Pareto set of solutions obtained from the space of design variables. In Fig. 3.6(b) and Fig. 3.7(b), Pareto front is consisted with objective function vector calculated from Pareto set of solutions in Fig. 3.6(a) and Fig. 3.7(a). In case of a single objective function, calculation was carried out on 7 cases of weighting factors defined by Eq. (3.24). However, to clarify the expression, only 3 cases were expressed.

When load W = 10 kN and operating speed n_s = 100 rps, the solutions for a single objective function by the EALA and for Pareto front and Pareto set of solutions by the PAL are compared. In case of α_1 = 1.0, it becomes a problem which only considers the temperature increase. The solutions and their corresponding temperature increases are located at the bottom right-hand corner for Pareto set of solutions and at the top left-hand corner for Pareto front as shown in Fig. 3.6. Also, in case of α_1 = 0, optimum problem is only considered for the supply flow. In Fig. 3.6, this is located at the other end of Pareto front and Pareto set of solutions. Lastly, in case where α_1 = 0.5, it is located in between the two cases described above for Pareto front and Pareto set of solutions. In other words, it can be confirmed that the problem of optimizing a single objective function in the form of linear combination is to find a specific solution to the corresponding Pareto optimization problem. From a different perspective, a designer can make selection according to the given situations by utilizing Pareto set of optimum solutions obtained.

Of the results obtained from PAL, value of the far-right solution in Fig. 3.7(a) is given by (λ, C) = (0.200, 247.3). It shows a difference of 52 μm or more in the upper limit of radial clearance of 300 μm. This result is related to temperature increase. When the operating speed n_s = 200 rps, load W = 10 kN and width to diameter ratio λ = 0.2, the temperature increase ΔT discontinuously rises for radial clearance exceeds 249.8 μm. As shown by Eqs. (3.12) and (3.18), this is due to the discontinuity of correction coefficients , α_m and G_θ^*, and

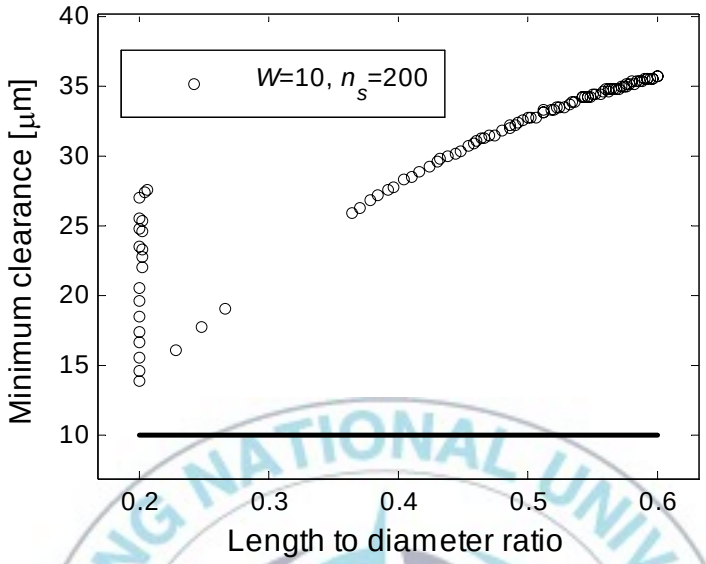
friction F_j according to Reynolds number.

As a result, in the above conditions, solutions exceeding C = 249.8 μm cannot be non-dominated solutions. Therefore, the domain below is excluded from the final Pareto set of solutions.

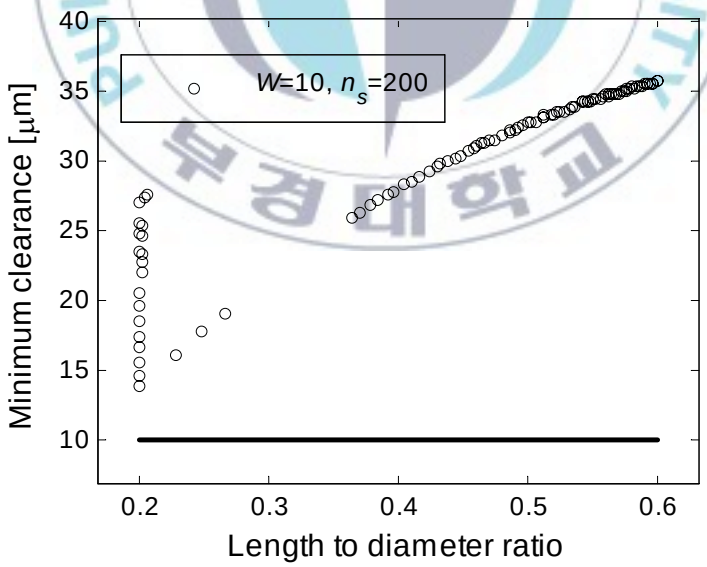
$$S_{ex} = \{(\lambda, C) \mid C \geq 249.8\ \mu\mathrm{m}\} \tag{3.25}$$

In Figs. 3.7(a) and (b), there are blanks in Pareto set of solutions and Pareto front and these are related to the constraints. Fig. 3.8 shows the constraints applicable in case W = 10 kN and n_s = 200 rps. The solid lines in Fig. 3.8 represent the limit values $(h_a, \Delta T_a, \omega, p_a)$. Although all other constraints are satisfied, whirling start at lower speed than operating speed as shown in Fig. 3.8(b). Therefore, a domain does not satisfy c_7 of Eq. (3.22) for constraints to exist. Also, this domain corresponds to the domain of discontinuity in Pareto set of solutions shown in Fig. 3.7(a). When W = 10 kN, all constraints, with the exception of upper and lower limits of design variables do not influence Pareto set of solutions in case where n_s = 100 or 150 rps.

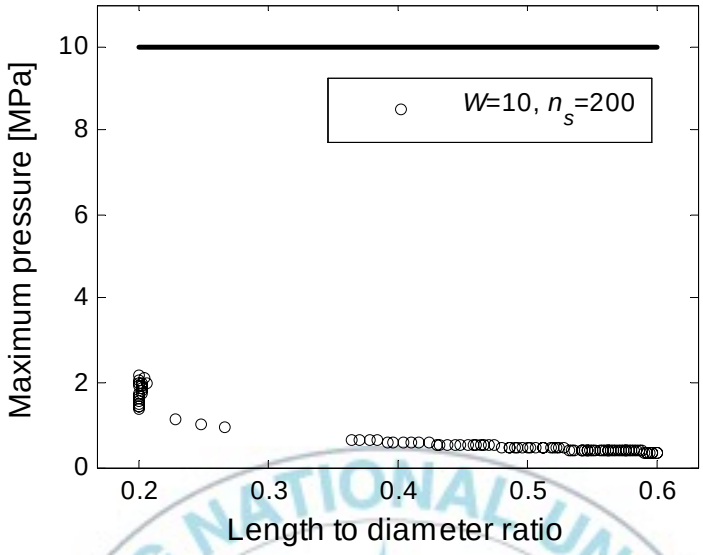
Fig. 3.9 shows Pareto set of solutions and Pareto front when the load is fixed at 10 kN and operating speeds change to 100, 200 and 300 rps. The arrows in Fig. 3.9(a) show the domain of radial clearance being excluded from Pareto set of solutions due to discontinuity of correction coefficient, as described previously. The six arrows in Fig. 3.9(b) point toward both ends of Pareto front. As predicted, it is confirmed that Pareto front moves in the direction of temperature increase as operating speed increases. At the same time, the expansion for Pareto front is also observed.



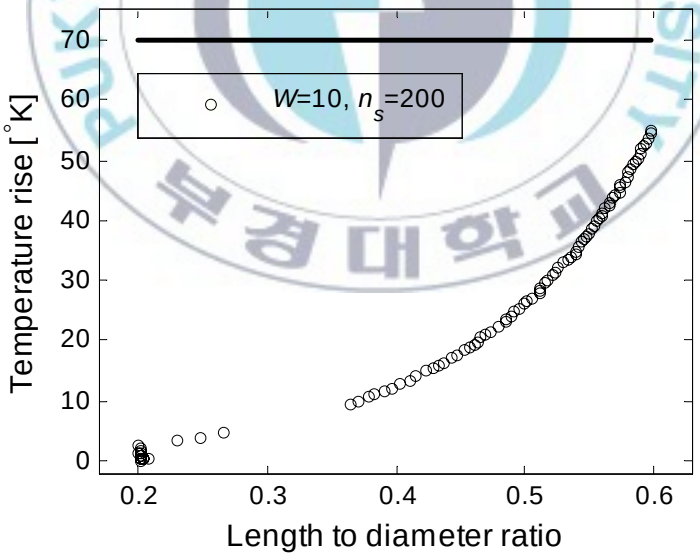
(a) Minimum Clearance to h_a



(b) Onset Speed to ω_{cr}

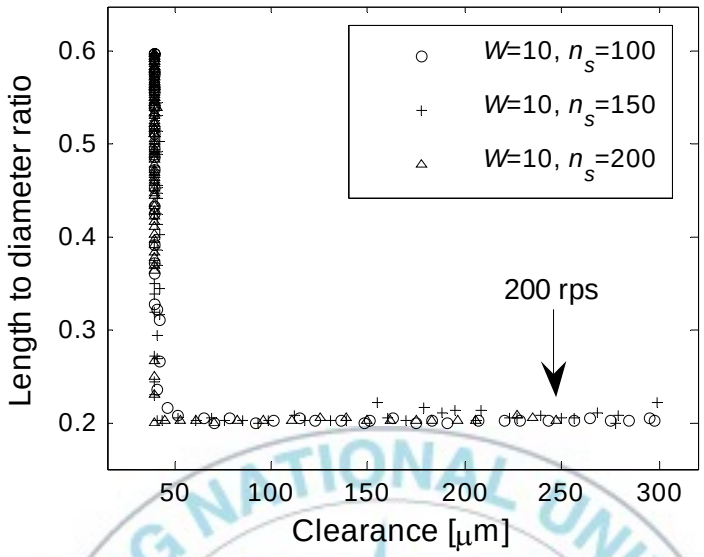


(c) Maximum Pressure to p_a

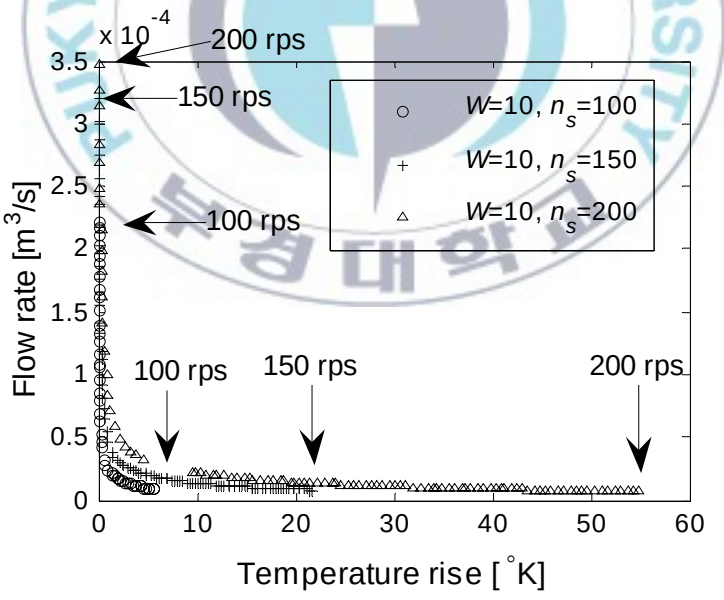


(d) Temperature Rise to ΔT_a

Fig. 3.8 Constraints: W = 10 kN, n_s = 200 rps.



(a) Pareto Optimal

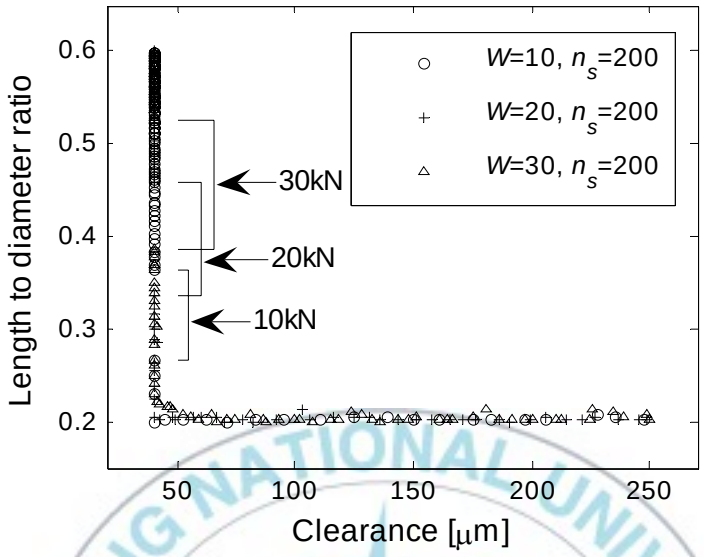


(b) Pareto Front

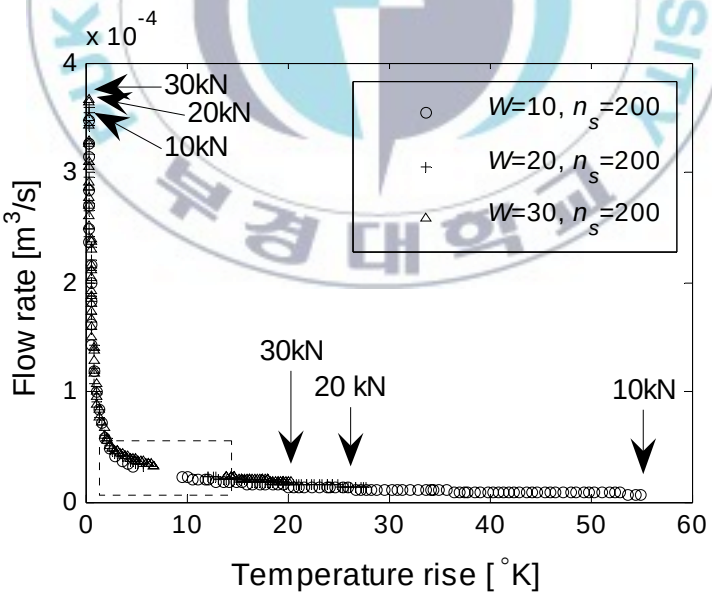
Fig. 3.9 Pareto Optimal and Pareto Front (W=10 kN, n_s=100, 150, 200 rps).

Fig. 3.10 shows the Pareto set of solutions and Pareto front in case where the load W is changed to 10, 20 and 30 kN, and operating speed fixes at n_s = 200 rps. For each case, the solutions of single objective functions in the form of linear combination are found and the results showed no noticeable discrepancy other than the contents of execution administered by changing the operation speed. Therefore, the result of single objective function is not fully explained. The domains marked with three arrows in Fig. 3.10(a) represent the domains excluded from Pareto set of solutions due to constraints under each load condition. The description for the operating condition of W = 10 kN and n_s = 200 rps is described previously. As the load increases, the domains excluded from the set of solutions start moving. Domains discontinuously broken by constraints in three sets of solutions from Pareto front as shown in Fig. 3.10(b) are expanded and presented in Fig. 3.10(c). In Fig. 3.11, the constraints for whirling starting speed are expressed according to each load. Fig. 3.8(b) is repeated in Fig. 3.11(a) for clarity. As shown in Fig. 3.8, design limits (ω_{cr}) are marked with solid lines. The three discontinuous domains in Fig. 3.10(a) correspond to the domains which do not satisfy the constraints as shown in the three inserts of Fig. 3.11.

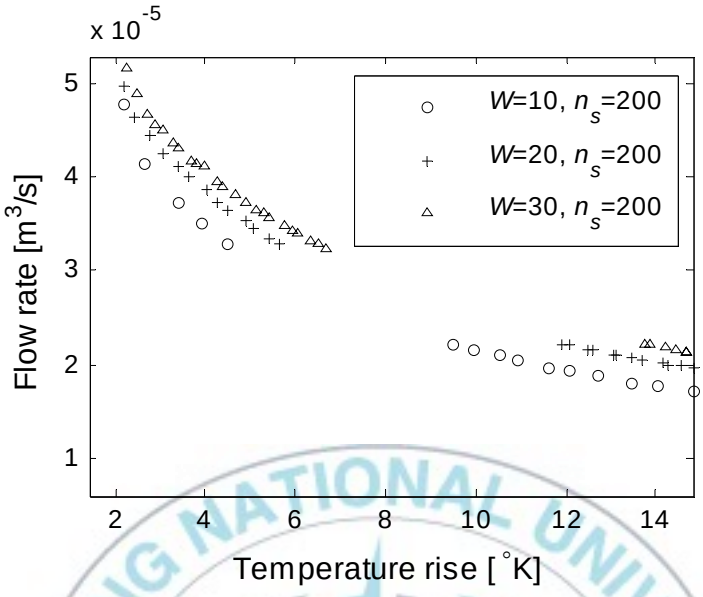
In all Pareto sets of solutions expressed in Fig. 3.10(a), S_{ex} of Eq. (3.25) is excluded. In other words, the exclusion from Pareto set of solutions due to discontinuity of correction coefficients and others occur in all cases. However, approximating the correction coefficients to eliminate this phenomenon will produce results which are not physically possible in reality. This phenomenon has been reflected with the physically existing discontinuity. Therefore, modification are not required if the equations represent well reality.



(a) Pareto Optimal

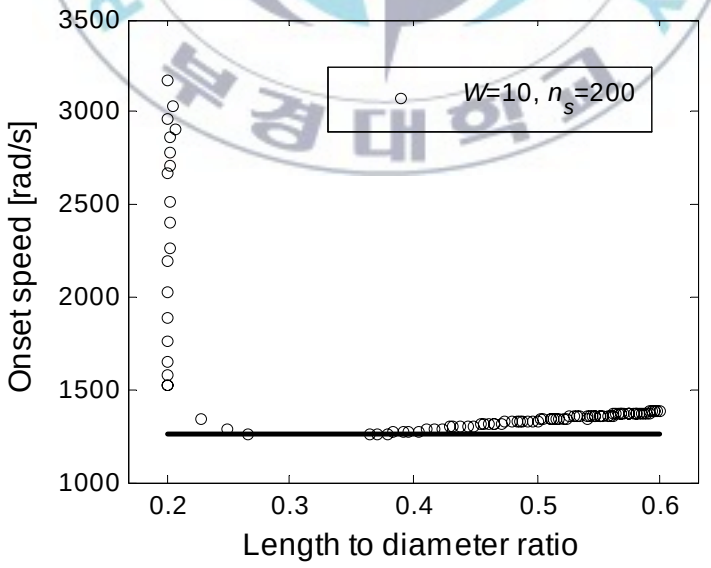


(b) Pareto Front

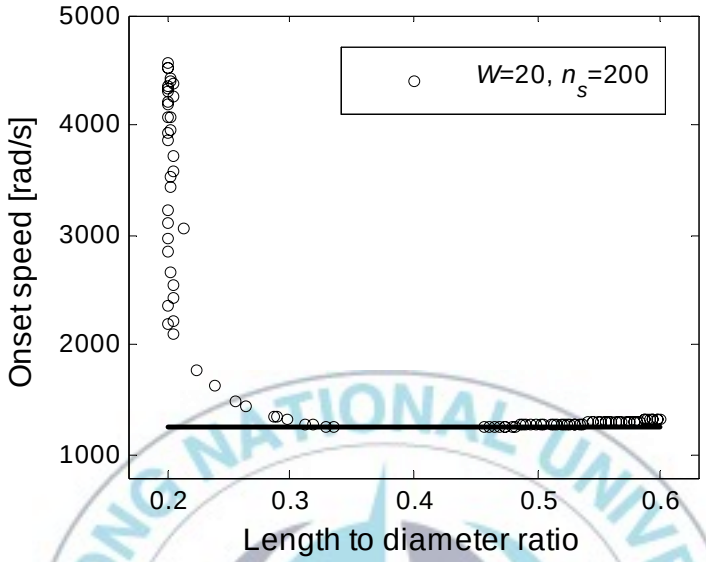


(c) Zoomed Pareto Front

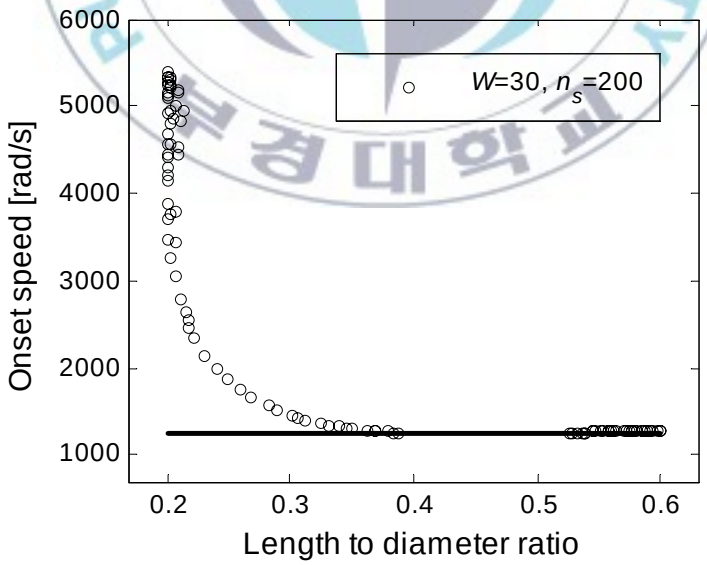
Fig. 3.10 Pareto Optimal and Pareto Front (W = 10, 20, 30 kN, n_s = 200 rps).



(a) W = 10 kN



(b) W = 20 kN



(c) W = 30 kN

Fig. 3.11 Onset Speed to ω_{cr} (n_s = 200 rps).

7. Conclusions

On the basis of artificial life optimization algorithm, Pareto artificial life optimization algorithm is proposed, which is capable of searching Pareto set of solutions of multiple objective functions. Through the three problems of multi-objective function optimization to hold known solutions, searching capabilities of the PAL on Pareto set of solutions has been verified. Also, by using the proposed algorithm, Pareto set of optimum solutions was found out by applying the PAL to the optimum design of the journal bearing. By comparing with the optimum solutions of single objective function, it has been confirmed in examples of this study that the solution of single objective function optimization is one of the specific cases of Pareto set of optimum solutions. In addition, the movement of Pareto front is confirmed when changing the conditions of load and operating speed. Lastly, it is confirmed that certain domains are excluded from Pareto set of solutions due to discontinuity of correction coefficient, etc. according to operating conditions.

References

[1] J. Anderson, 2000, A Survey of Multiobjective Optimization in Engineering Design, Technical Report LiTH-IKP-R-1097, Linkoping University, Sweden

[2] J.D. Schaffer, 1985, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, 160-168.

[3] S.J. David, 1985, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, 93-100.

[4] D.E. Goldberg, 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc.

[5] J. Horn, N. Nafpliotis, D.E. Goldberg, 1994, "A Niched Pareto Genetic Algorithm for Multiobjective Optimization," Proceeding of the 5th International Conference on Genetic Algorithms, 82-87.

[6] N. Srinivas, K. Deb, 1995, "Multi-objective Optimization Using Nondominated Sorting in Genetic Algorithms," Evolutionary Computation, 2(3), 221-248.

[7] Y. Shiyou, G. Ni, 1998, "An Universal Tabu Search Algorithm for Global Optimization of Multimodal Functions with Continuous Variables in Electromagnetics," IEEE Trans. Magnetics, 34, 2901-2904.

[8] S.L. Ho, S. Yang, G. Ni, H.C. Wong, 2002, "A Tabu Method to Find the Pareto Solutions of Multiobjective Optimal Design Problems in Electro-magnetics," IEEE Trans. Magnetics, 38(2), 1013-1016.

[9] B.S. Yang, Y.H. Lee, 2000, "Artificial Life Algorithm for Function Optimization," ASME Design Engineering Technical Conferences DETC 2000/ DAC-1524.

[10] B.S. Yang, Y.H. Lee, B.K. Choi, H.J. Kim, 2001, "Optimum Design of

Short Journal Bearings by Artificial Life Algorithm," Tribology International, 34(7), 427-435.

[11] B.S. Yang, J.D. Song, 2002, "Development of an Enhanced Artificial Life Optimization Algorithm and Optimum Design of Short Journal Bearings," Trans. Korean Society of Noise and Vibration Engineering, 12(6), 478-487.

[12] J.D. Song, B.S. Yang, B.K. Choi, H.J. Kim, 2005, "Optimum Design of Short Journal Bearings by Enhanced Artificial Life Optimization Algorithm," Tribology International, 38, 403-412.

[13] Y.K. Ahn, J.D. Song, B.S. Yang, 2003, "Optimal Design of Engine Mount Using an Artificial Life Algorithm," Journal of Sound and Vibration, 261(1), 309-328.

[14] Y.K. Ahn, J.D. Song, B.S. Yang, K.K. Ahn, S. Morishita, 2005, "Optimal Design of Nonlinear Fluid Engine Mount," Journal of Mechanical Science and Technology, 19(3), 768-777.

[15] C.A.C. Coello, C.S.P. Zacatenco, 2006, "Twenty Years of Evolutionary Multi-objective Optimization: a Historical View of the Field," Departamento de Ingenier´ıa El´ectrica, Secci´on de Computaci´on, Av. Instituto Polit´ecnico Nacional No. 2508.

[16] V. Pareto, 1896, "Cours D'economie Politique," Vol. I and II, F. Rouge, Lausanne.

[17] C.M. Fonseca, P.J. Fleming, 1995, "Multi-objective Genetic Algorithms Made Easy: Selection, Sharing, and Mating Restriction," Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, 45-52.

[18] M. Tanaka, 1995, "GA-based Decision Support System for Multi-criteria Optimization," Proceedings of the International Conference on Systems, Man and Cybernetics-2, 1556-1561.

[19] H. Hashimoto, 1997, "Optimum Design of High-speed Short Journal

Bearings by Mathematical Programming," Tribology Transactions, 40, 283-293.

Ⅳ. Conclusions

The parameters of the EALA have been simplified. The effectiveness of the EALA was evaluated using three test functions. The optimized results were compared with those of conventional artificial life algorithm. The results show that the EALA reaches the optimum solution faster and closer to exact solution than the conventional artificial life algorithm, and will give all global optimum solutions in a solution space. The algorithm reduced the time needed for searching, to say nothing of accuracy of optimum solutions, because the ELAL improved the method of making decision of new points for offspring and for own waste by using random tabu search method.

On the basis of the artificial life optimization algorithm, Pareto artificial life optimization algorithm is proposed, which is capable of searching Pareto set of solutions of multiple objective functions. Through the three problems of multi-objective function optimization to hold known solutions, searching capabilities of PAL on Pareto set of solutions was verified. By using the proposed algorithm, Pareto set of optimum solutions was found out by applying PAL to the optimum design of the journal bearing. Also the EALA was applied to the same problem. By comparing Pareto set of optimum solutions with the optimum solutions of single objective function, it was confirmed in examples of this study that the solution of single objective function optimization is one of the specific cases of Pareto set of optimum solutions. In addition, the movement of Pareto front was confirmed when changing the conditions of load and operating speed. Lastly, it was confirmed that certain domains were excluded from Pareto set of solutions due to discontinuity of correction coefficient, etc. according to operating conditions.

다목적 최적화 문제를 위한 파레토인공생명 최적화 알고리듬

송 진 대

부경대학교 대학원 기계공학과

국문요약

많은 공학적 최적화 문제는 단일 목적 함수보다는 다수의 목적 함수로 구성되는 경우가 많다. 이러한 다목적 최적화 문제는 여러 목적함수를 하나의 목적함수로 변환하고 통상의 단일 목적함수 방법을 적용하여 푸는 방법과, 2 개 이상의 목적함수를 동시에 고려하여 파레토 해집합이라 불리는 최적해 집합을 구하는 방법으로 대별된다.

전자는 일반적으로 다수의 목적 함수를 선형 조합의 형태 혹은 곱의 형태를 취하여 단일 목적함수로 치환하는 방법을 사용하거나, 중요한 하나의 목적 함수만을 설정하고 다른 목적 함수들은 구속조건으로 처리하는 방법을 택한다. 이러한 방법들은 최적화 과정에서 각 후보해들을 비교하는 단일 기준을 확립하려는 방법론적인 원인과 하나의 최적해(혹은 근사 최적해)를 도출하고자 하는 목적에서 시작되었다. 다른 방법으로 다수의 목적 함수를 동시에 목적 함수로 고려하는 방법을 파레토 최적화(Pareto optimization)라 한다. 이는 최종 의사결정자에게 가능한 해를 제공하려는 목적으로 단일 목적함수로 변환하여 하나의 해만을 얻게 됨으로써 존재하는 다른 가능성을 찾지

못하는 단점을 보완하고자 하는 것이다. 또 다른 방법으로는 문제를 목표 계획법으로 변환하여 공식화하기도 한다.

함수 최적화의 경우, 휴리스틱 최적화 방법들은 문제에 대한 특별한 제한사항을 가지고 있지 않는 장점을 가지고 있고, 전역 최적해에 대한 탐색능력이 우수한 것으로 평가되고 있다. 특히 휴리스틱 전역 최적화 기법의 하나로 함수최적화를 위한 인공생명 알고리듬이 개발되어 저널 베어링, 엔진 마운트 등의 최적 설계 문제를 통해 그 유용성이 입증되었다.

인공생명이란 용어는 1987 년 Langton 이 주관한 제 1 회 인공생명 workshop 에서부터 시작되었다. Langton 은 “자연계의 살아있는 시스템이 나타내는 거동을 모의하는 인공시스템에 대하여 연구하는 분야”로 인공생명을 정의하였다. 인공생명의 기본적인 동기는 지구상에 존재하는 자연적인 생명체의 모델을 넘어선 가능한 생명체를 합성함으로써 더욱 넓은 생명의 영역을 탐구하고, 생명의 진정한 의미를 파악하는 데 있다. 인공생명 최적화 알고리듬은 생물군의 군집형성을 모의함으로써 최적화를 실현하는 최적화 방법론이다.

본 연구에서는 인공생명 알고리듬에 배분적응도의 개념과 파레토 목록(Pareto archive)을 도입하여 파레토 최적화 문제에 적용 가능하도록 확장하였다. 이를 통해 다양한 공학적 문제를 실용적인 관점에서 해결할 수 있도록 하였다.

제안된 방법론으로 저널 베어링의 최적 설계에 적용하여 하중 및 운전속도 등의 파라미터에 따른 파레토 최적해를 구하고 그 유용성을 확인하였다. 또, 단일목적함수로 변환된 단일목적함수 최적화 문제의 해와 비교를 하여, 선형조합 형태의 단일 목적 함수 최적화 문제와 파레토 최적화 문제와의 관련성을 검토하였다. 그 결과로 선형조합된 단일목적함수의 최적해는 파레토 최적화 문제의 파레토 해집합의 특수한 경우임을 확인하였다.

Acknowledgements

실험실에 들어온 이후로 많은 시간이 지나갔습니다. 스스로 만족스러운 시간도 있었으며 그러하지 못한 시간도 있었으나 모두 의미 있는 시간이었다고 생각합니다.

학위논문을 마무리 하는 과정까지 양보석 교수님의 학문적인 그리고 정신적인 충고와 모범은 저에게 특별한 가치를 지니고 있으며, 앞으로의 제 삶에도 유의미한 가치로 남아 있을 것입니다. 부족한 논문의 사사를 통해 고마움을 전합니다.

학위논문을 마지막까지 검토해주셨던 이수종 교수님, 김병탁 교수님, 김선진 교수님, 그리고 최병근 박사님의 수고에 감사드립니다.

학위과정에서 여러 선배님들, 후배님들과 생활을 하였습니다. 지금 그 분들을 모두 여기에 나열하려니 너무 많군요! 이제는 모두 졸업하시고 대학과 기업에서 열심히 일하고 계시는 여러 선배님들께 많은 배움을 얻었습니다. 짧은 글로 고마움을 전합니다. 그리고 졸업하고 직장에서 일하고 있거나 혹은 아직 학교에 남아 있는 후배님들의 도움을 많이 받았던 것을 기억하고 있으며, 고마워 하고 있습니다.

많은 시간을 아무 말 없이 바라봐 주신 가족들에게 특별한 고마움을 전합니다. 큰 형님과 작은 형님 그리고 형수님 에게 죄송함과 감사함을 전합니다. 그리고 누님과 매형도 계속 행복하기를 바랍니다. 마지막으로 아버님과 어머님께 감사를 드립니다.

모두들 감사한 분들입니다. 잊지 않고 살아가겠습니다.