We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Particle Swarm Optimization with External Archives for Interactive Fuzzy Multiobjective Nonlinear Programming

Takeshi Matsui, Masatoshi Sakawa, Kosuke Kato and Koichi Tamada
Hiroshima University
Japan

1. Introduction

In general nonlinear programming problems to find a solution which minimizes an objective function under given constraints, one whose objective function and constraint region are convex is called a convex programming problem. For such convex programming problems, there have been proposed many efficient solution method as the successive quadratic programming method and the general gradient method. Unfortunately, there have not been proposed any decisive solution method for nonconvex programming problems. As practical solution methods, meta-heuristic optimization methods as the simulated annealing method and the genetic algorithm have been proposed.

In recent years, however, more speedy and more accurate optimization methods have been desired because the size of actual problems has been increasing.

As a new optimization method, particle swarm optimization (PSO) was proposed (Kennedy & Eberhart, 1995). PSO is a search method simulating the social behavior that each individual in the population acts by using both the knowledge owned by it and that owned by the population, and they search better points by constituting the population. The authors proposed a revised PSO (rPSO) by incorporating the homomorphous mapping and the multiple stretching technique in order to deal with shortcomings of the original PSO as the concentration to local solution and the inapplicability of constrained problems (Matsui et al., 2008).

In recent years, with the diversification of social requirements, the demand for the programs with multiple objective functions, which may be conflicting with each other, rather than a single-objective function, has been increasing (e.g. maximizing the total profit and minimizing the amount of pollution in a production planning). Since there does not always exist a complete optimal solution which optimizes all objectives simultaneously for multiobjective programming problems, the Pareto optimal solution or non-inferior solution, is defined, where a solution is Pareto optimal if any improvement of one objective function can be achieved only at the expense of at least one of the other objective functions. For such multiobjective optimization problems, fuzzy programming approaches (e.g. (Zimmermann, 1983), (Rommelfanger, 1996), considering the imprecise nature of the DM's judgments in multiobjective optimization problems, seem to be very applicable and promising. In the application of the fuzzy set theory into multiobjective linear programming problems started

(Zimmermann, 1978), it has been implicitly assumed that the fuzzy decision or the minimum-operator of (Bellman & Zadeh ,1970) is the proper representation of the DM's fuzzy preferences. Thereby, M. Sakawa et al. have proposed interactive fuzzy satisficing methods to derive satisficing solutions for the decision maker along with checking the local preference of the decision maker through interactions for various multiobjective programming problems (Sakawa et al, 2002).

In this paper, focusing on multiobjective nonlinear programming problems, we attempt to derive satisficing solutions through the interactive fuzzy satisficing method. Since problems solved in the interactive fuzzy satificing method for multiobjective nonlinear programming problems are nonlinear programming problems, we adopt rPSO (Matsui et al, 2008) as solution methods to them. In particular, we consider measures to improve the performance of rPSO in applying it to solving the augmented minimax problem.

2. Multiobjective nonlinear programming problems

In this paper, we consider multiobjective nonlinear programming problem as follows:

$$\text{minimize} \qquad f_l(\boldsymbol{x}), \; l=1, 2, \ldots, k$$

$$\text{subject to} \qquad g_i(\boldsymbol{x}) \leq 0, \; i=1, 2, \ldots, m \qquad (1)$$

$$l_j \leq x_j \leq u_j, \; j=1, 2, \ldots, n$$

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T \in R^n$$

where $f_l(\cdot)$, $g_i(\cdot)$ are linear or nonlinear functions, l_j and u_j are the lower limit and the upper limit of each decision variable x_j. In addition, we denote the feasible region of (1) by X.

3. An interactive fuzzy satisficing method

In order to consider the imprecise nature of the decision maker's judgments for each objective function in (1), if we introduce the fuzzy goals such as ``$f_l(\boldsymbol{x})$ should be substantially less than or equal to a certain value'', (1) can be rewritten as:

$$\begin{array}{c} \text{maximize} \\ \boldsymbol{x} \in X \end{array} \quad (\mu_1(f_1(\boldsymbol{x})), \ldots, \mu_k(f_k(\boldsymbol{x}))) \qquad (2)$$

where $\mu_l(\cdot)$ is the membership function to quantify the fuzzy goal for the l th objective function in (1).

Since (2) is regarded as a fuzzy multiobjective decision making problem, there rarely exists a complete optimal solution that simultaneously optimizes all objective functions. As a reasonable solution concept for the fuzzy multiobjective decision making problem, M. Sakawa defined M-Pareto optimality on the basis of membership function values by directly extending the Pareto optimality in the ordinary multiobjective programming problem (Sakawa, 1993). In the interactive fuzzy satisficing method, in order to generate a candidate for the satisficing solution which is also M-Pareto optimal, the decision maker is asked to specify the aspiration levels of achievement for all membership functions, called the reference membership levels (Sakawa, 1993). For the decision maker's reference membership

levels $\bar{\mu}_l$, $l=1, \ldots, k$, the corresponding M-Pareto optimal solution, which is nearest to the

requirements in the minimax sense or better than it if the reference membership levels are attainable, is obtained by solving the following augmented minimax problem (3).

$$\underset{x \in X}{\text{minimize}} \quad \underset{l=1, \dots, k}{\max} \quad \left\{ (\bar{\mu}_l - \mu_l(f_l(x))) + \rho \sum_{i=1}^{k} (\bar{\mu}_i - \mu_i(f_i(x))) \right\} \qquad (3)$$

where ρ is a sufficiently small positive number.

We can now construct the interactive algorithm in order to derive the satisficing solution for the decision maker from the M-Pareto optimal solution set. The procedure of an interactive fuzzy satisficing method is summarized as follows.

Step 1:

Under a given constraint, minimal value and maximum one of each objective function are calculated by solving following problems.

$$\underset{x \in X}{\text{minimize}} \quad f_l(x), l=1, 2, \dots, k \qquad (4)$$

$$\underset{x \in X}{\text{maximize}} \quad f_l(x), l=1, 2, \dots, k \qquad (5)$$

Step 2:

In consideration of individual minimal value and maximum one of each objective function, the decision maker subjectively specifies membership functions $\mu_l(f_l(x))$, $l=1, \dots, k$ to quantify fuzzy goals for objective functions. Next, the decision maker sets initial reference membership function values $\bar{\mu}_l$, $l=1, \dots, k$.

Step 3:

We solve the following augmented minimax problem corresponding to current reference membership function values (3).

Step 4:

If the decision maker is satisfied with the solution obtained in Step 3, the interactive procedure is finished. Otherwise, the decision maker updates reference membership function values $\bar{\mu}_l$, $l=1, 2, \dots, k$ based on current membership function values and objective function values, and return to Step 3.

4. Particle swarm optimization

Particle swarm optimization (Kennedy & Eberhart, 1995) is based on the social behavior that a population of individuals adapts to its environment by returning to promising regions that were previously discovered (Kennedy & Spears, 1998). This adaptation to the environment is a stochastic process that depends on both the memory of each individual, called particle, and the knowledge gained by the population, called swarm. In the numerical implementation of this simplified social model, each particle has four attributes: the position vector in the search space, the velocity vector and the best position in its track and the best position of the swarm. The process can be outlined as follows.

Step 1:

Generate the initial swarm involving N particles at random.

Step 2:
    Calculate the new velocity vector of each particle, based on its attributes.
Step 3:
    Calculate the new position of each particle from the current positon and its new velocity vector.
Step 4:
    If the termination condition is satisfied, stop. Otherwise, go to Step 2.

To be more specific, the new velocity vector of the i-th particle at time t, $\boldsymbol{v}_i^{t+1}$ is calculated by the following scheme introduced by (Shi & Eberhart, 1998).

$$\boldsymbol{v}_i^{t+1} := \omega^t \boldsymbol{v}_i^t + c_1 R_1^t (\boldsymbol{p}_i^t - \boldsymbol{x}_i^t) + c_2 R_2^t (\boldsymbol{p}_g^t - \boldsymbol{x}_i^t) \qquad (6)$$

In (6), R_1^t and R_2^t are random numbers between 0 and 1, p_i^t is the best position of the i-th particle in its track and p_g^t is the best position of the swarm. There are three problem dependent parameters, the inertia of the particle ω^t, and two trust parameters c_1, c_2. Then, the new position of the i-th particle at time t, x_i^{t+1}, is calculated from (7).

$$x_i^{t+1} := x_i^t + v_i^{t+1} \qquad (7)$$

where x_i^t is the current position of the i-th particle at time t. The i-th particle calculates the next search direction vector $\boldsymbol{v}_i^{t+1}$ by (6) in consideration of the current search direction vector $\boldsymbol{v}_i^t$, the direction vector going from the current search position x_i^t to the best position in its track p_i^t and the direction vector going from the current search position x_i^t to the best position of the swarm p_g^t, moves from the current position x_i^t to the next search position x_i^{t+1} calculated by (7). The parameter ω^t controls the amount of the move to search globally in early stage and to search localy by decreasing ω^t gradually.
The searching procedure of PSO is shown in Fig. 1.

Comparing the evaluation value of a particle after movement, $f(\boldsymbol{x}_i^{t+1})$, with that of the best position in its track, $f(p_i^t)$, if $f(\boldsymbol{x}_i^{t+1})$ is better than $f(p_i^t)$, then the best position in its track is updated as $p_i^t := x_i^{t+1}$. Futhermore, if $f(p_i^{t+1})$ is better than $f(p_g^t)$, then the best position in the swarm is updated as $p_g^{t+1} := p_i^{t+1}$.
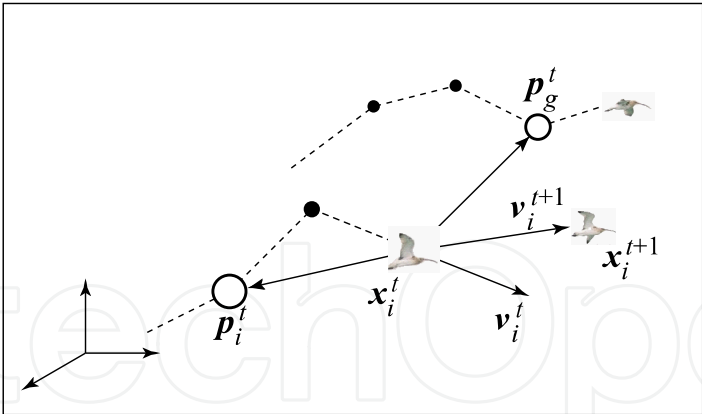
Figure 1. Movement of a particle in PSO

In the original PSO method, however, there are drawbacks that it is not directly applicable to constrained problems and it is liable to stopping around local optimal solutions.

To deal with these drawbacks of the original PSO method, we incorporate the bisection method and a homomorphous mapping to carry out the search considering constraints.

In addition, we proposed the multiple stretching technique and modified move schemes of particles to restrain the stopping around local optimal solutions (Matsui et al., 2008).

Thus, we applied rPSO for interactive fuzzy multiobjective nonlinear programming problems and proposed multiobjective revised PSO (MOrPSO) method incorporating move scheme to the nondominated particle in order to search effectively for the augmented minimax problmes (Matsui et al., 2007). In the application of large-scale augmented minimax problem, MOrPSO method is superior than rPSO method on efficiency. On the other hand, MOrPSO method is inferior on accuracy.

5. Improvement of MOrPSO

We show the results of the applicaltion of the original rPSO (Matsui et al., 2008) and MOrPSO (Matsui et al., 2007) to the augmented minimax problem for multiobjective nonlinear programming problem with $l = 2$, $n = 55$ and $m = 100$ in Table 1. In these experiments we set the swarm size $N = 70$, the maximal search generation number $T_{\max} =$ 5000. In addition, we use the following membership functions: $\bar{\mu}_1 = 1.0$, $\bar{\mu}_2 = 1.0$.

method	objective function value (minimize)			computational time (sec)
	best	average	worst	
rPSO	0.3464	0.4471	0.5632	144.45
MOrPSO	0.3614	0.4095	0.4526	129.17

Table 1. Results of the application to the augmented minimax problem

From Table 1, MOrPSO method is superior than rPSO method on efficiency in the average value, the worst one and computational time. However, the best value of MOrPSO method is worse than that of rPSO method, MOrPSO method is inferior on accuracy. We consider

the case that the search accuracy turns worse incorporating the direction to nondominated particle (approximate M-Pareto optimal solution) in MOrPSO method.

In this paper, we improve the search accuracy incorporating external archives to record nondominated particles in the swarm. Here, as recorded nondominated particle increases in archives, computational time increases in order to judge whether a particle is nondominated.

Therefore, there is many computational time that we record all the nondominated particles to archives. Thus we divide membership function space with hypercube shown in Fig. 2 and record a number of nondominated particle included in each hypercube.
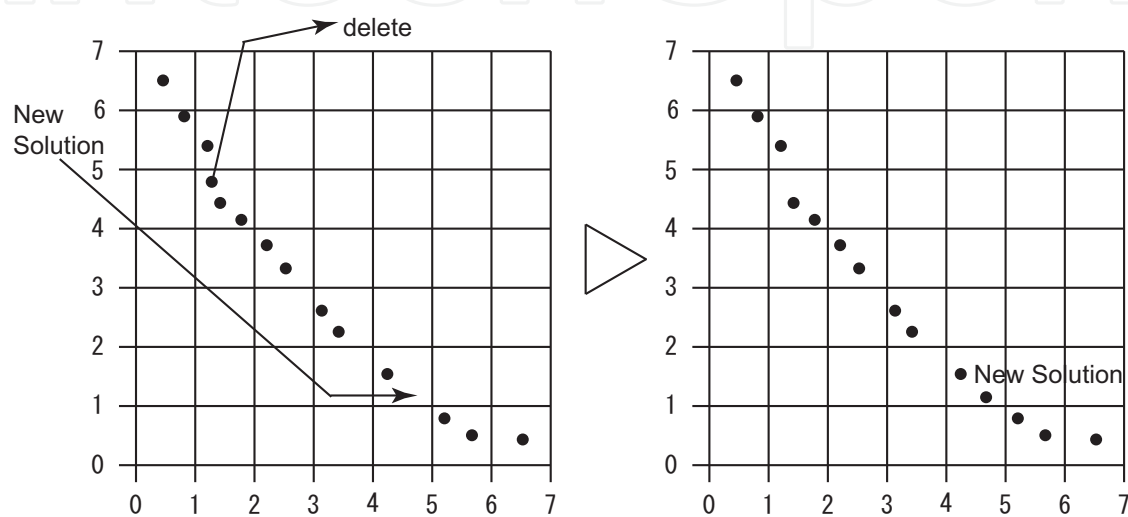

Figure 2. reduction of archives with grid (l=2)

When a number of nondominated particle recorded in archives is greater than a fixed number, we delete one particle from hypercube with many numbers of nondominated particle and record new solution (particle). We consider that can reduce computational time and express approximate M-Pareto optimal front by a few particles incorporating reduction of archives. We show the results of the application of MOrPSO method incorporating reduction of archives (MOrPSO-1) to the above same problem in Table 2.

method	objective function value (minimize)			computational time (sec)
	best	average	worst	
MOrPSO-1	0.4379	0.4708	0.5030	166.78

Table 2. Results of the application to the augmented minimax problem

From Table 2, it is clear that all of the best, average, worst value and computational time obtained by MOrPSO-1 are worse than those obtained by MOrPSO. We consider that a particle moves using nondominated particle which is not useful since all nondominated particles in archives are used in search. Thus, in the membership function space, we consider that information of a particle exsiting far from the reference membership value is hard to contribute to search and introduce threshold value for selection of nondominated particle as shown in Fig. 3.
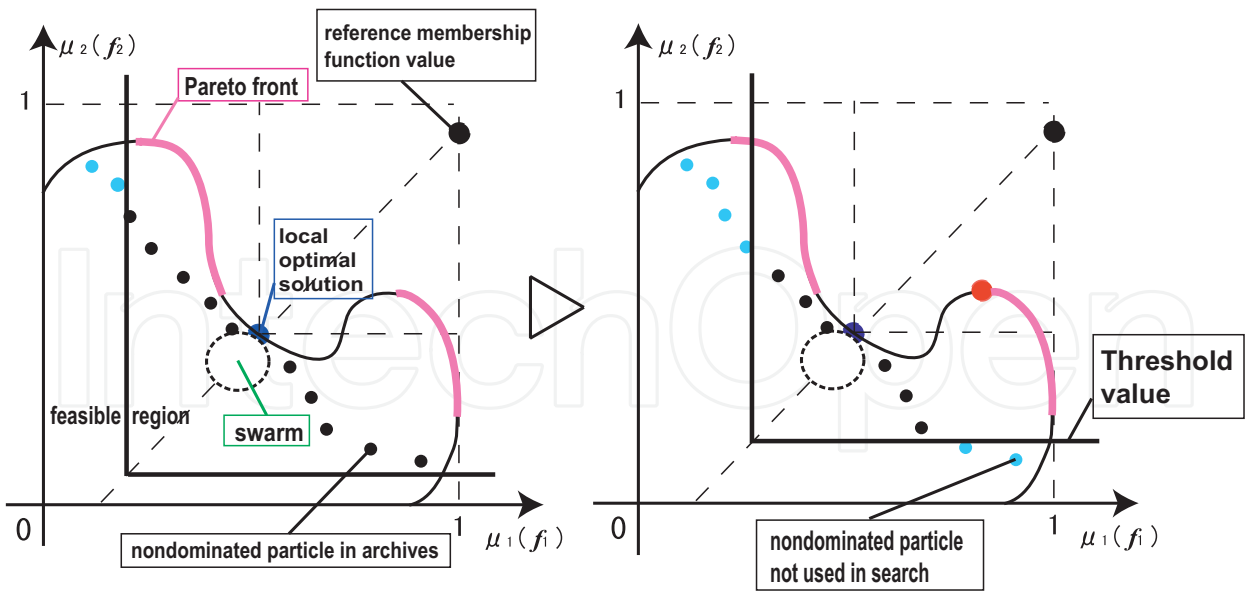
Figure 3. Limit of nondominated particle by threshold value (*l*=2)

We show the results of the application of MOrPSO method incorporating limitation by
threshold value (MOrPSO-2) to the above same problem in Table 3.

method	objective function value (minimize)			computational time (sec)
	best	average	worst	
MOrPSO-2	0.2993	0.3430	0.3777	165.71

Table 3. Results of the application to the augmented minimax problem

From Table 3, in the application of MOrPSO method incorporating limitation by threshold
value (MOrPSO-2), we can get better solutions in the sense of best, average and worst than
those obtained by rPSO and MOrPSO.

In order to show the efficiency of the proposed PSO, we consider the multiobjective
nonlinear programming problem with $l = 2$ and $n = 100$. In these experiments, we set the
swarm size $N = 100$, the maximal search generation number $T_{max} = 5000$. In addition, we use

the following reference membership function values: $\bar{\mu}_1 = 1.0$, $\bar{\mu}_2 = 1.0$.

method	objective function value (minimize)			computational time (sec)
	best	average	worst	
rPSO [6]	0.2547	0.2783	0.3251	26.91
MOrPSO [7]	0.1950	0.2033	0.2208	32.58
MOrPSO-2	0.2018	0.2320	0.2711	28.11

Table 4. Results of the application to the augmented minimax problem

From Table 4, it is clear that all of the best, average, worst value and computational time
obtained by MOrPSO-2 are worse than those obtained by MOrPSO. We consider that it

occurs to make no use of the information of nondominiated particle in search since there is a few nondominiated particle information stored in archives of MOrPSO-2 and only the best value of each objective function and the best value limb of the augmented minimax problem are saved. Therefore, we propose MOrPSO with external archives (MOrPSO-EA) using nondominated particle in the swarm same as MOrPSO in order to store various nondominated particle as possible in normal search. And the results of the application are shown in Table 5.

method	objective function value (minimize)			computational time (sec)
	best	average	worst	
MOrPSO	0.1950	0.2033	0.2208	32.58
MOrPSO-EA	0.1746	0.1787	0.1842	29.01

Table 5. Results of the application to the augmented minimax problem

From Table 5, in the application of MOrPSO-EA, we can get better solutions in the sense of best, average and worst than those obtained by MOrPSO.

6. Numerical examples

In MOrPSO, it searches globaly in the early generation and localy decreasing ω^t. However, we consider that necessity to search globaly in the early generation is low after the second time since the information of nondominated particle to current generation is stored in archives in proposed MOrPSO. Therefore, we consider that the proposed MOrPSO can search localy in the early generation.

In order to show the efficiency of the proposed MOrPSO, we consider the multiobjective nonlinear programming problem with $l = 2$ and $n = 100$ and $m = 55$. In these experiments, we set the maximal search generation number of MOrPSO and the proposed MOrPSO (MOrPSO-EA) in the 1st interactive T_{max} = 5000 and in the 2nd and 3rd interactive T_{max} = 3000. We show the results of the application are shown in Table 6 and 7.

interactive	1st	2nd	3rd
$\bar{\mu}_1$	1.0	1.0	0.85
$\bar{\mu}_2$	1.0	0.7	0.7
$\mu_1(x)$	0.6157	0.8003	0.7575
$\mu_2(x)$	0.6157	0.5003	0.6075
minimax value	0.3844	0.1997	0.0925
time (sec)	127.20	128.89	131.25

Table 6. Interactive fuzzy programming through MOrPSO

interactive	1st	2nd	3rd
$\bar{\mu}_1$	1.0	1.0	0.85
$\bar{\mu}_2$	1.0	0.7	0.7
$\mu_1(x)$	0.7183	0.8458	0.8042
$\mu_2(x)$	0.7183	0.5458	0.6542
minimax value	0.2817	0.1542	0.0458
time (sec)	157.37	98.58	101.16

Table 7. Interactive fuzzy programming through MOrPSO-EA (proposed)

From Table 6 and 7, MOrPSO-EA is superior than MOrPSO on accuracy. In addition, we can decrease total computational time by reducing the maximal search generation number.

6. Conclusion

In this research, we focused on multiobjective nonlinear programming problems and proposed a new MOrPSO technique which is accuracy for in applying the interactive fuzzy satisficing method. In particular, considering the features of augmented minimax problems solved in the interactive fuzzy satisficing method, we incorporated use of external archives, reduction of archives and the limitation of threshold value. Finally, we showed the efficiency of the proposed MOrPSO by applying it to numerical examples.

7. References

R.E. Bellman & L.A. Zadeh. (1970). Decision making in a fuzzy environment, *Management Science*, Vol. 17, pp. 141-164

C.A.C Coello, G.T. Pulido, M.S. Lechuga. (2004). Handling multiple objectives with particle swarm optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 256-279

V.L. Huang, A.K. Qin, K. Deb, E. Zitzler, P.N. Suganthan, J.J. Liang, M. Preuss, S. Huband. (2007). Problem definitions for performance assessment on multi-objective optimization algorithms, *http://www.ntu.edu. sg/home/EPNSugan/index-files/CEC-07/CEC-07-TR-13-Feb.pdf*

J. Kennedy & R.C. Eberhart. (1995). Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942-1948

T. Matsui, K. Kato, M. Sakawa, T. Uno, K. Morihara. (2008). Nonlinear Programming Based on Particle Swarm Optimization, In: *Advances in Industrial Engineering and Operations Research*, Alan H. S. Chan, Sio-Iong Ao (Ed.), pp. 173-183, Springer, New York, 2008.

T. Matsui, M. Sakawa, K. Kato, T. Uno, K. Tamada. (2007). An interactive fuzzy satisficing method through particle swarm optimization for multiobjective nonlinear programming problems, *Proceedings of the 2007 IEEE Symposium Series on Computational Intelligence*, pp. 71-76

H. Rommelfanger. (1996). Fuzzy linear programming and applications, *European Journal of Operational Research*, vol. 92, pp. 512-528

M. Sakawa. (1993). *Fuzzy Sets and Interactive Multiobjective Optimization*, Plenum Press, New York

M. Sakawa, K. Kato, T. Suzuki. (2002). An interactvie fuzzy satisficing method for multiobjective non-convex programming problems through genetic algorithms, *Proceedings of 8th Japan Society for Fuzzy Theory and Systems Chugoku / Shikoku Branch Office Meeting*, pp. 33-36

Y.H. Shi & R.C. Eberhart. (1998). A modified particle swarm optimizer, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69-73

H.-J. Zimmermann. (1978). Fuzzy programming and linear programming with several objective functions, *Fuzzy Sets and Systems*, Vol. 1 pp. 45-55

H.-J. Zimmermann. (1983). Fuzzy mathematical programming, *Computers & Operations Research*, Vol. 10, pp. 291-298
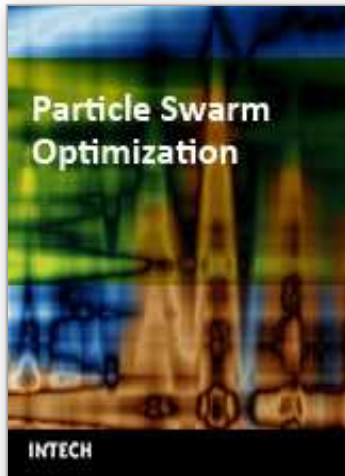
Particle Swarm Optimization

Edited by Aleksandar Lazinica

Particle swarm optimization (PSO) is a population based stochastic optimization technique influenced by the social behavior of bird flocking or fish schooling.PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Takeshi Matsui, Masatoshi Sakawa, Kosuke Kato and Koichi Tamada (2009). Particle Swarm Optimization with External Archives for Interactive Fuzzy Multiobjective Nonlinear Programming, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0, InTech, Available from:
http://www.intechopen.com/books/particle_swarm_optimization/particle_swarm_optimization_with_external_ar chives_for_interactive_fuzzy_multiobjective_nonlinear_pr

INTECH
open science | open minds