Influence of Fitness Quantization Noise on the Performance of Interactive PSO

Nakano, Yu
Graduate School of Design, Kyushu University

Takagi, Hideyuki
Faculty of Design, Kyushu University

出版情報：IEEE Congress on Evolutionary Computation (CEC). 2009, pp.2146-2422, 2009. IEEE
バージョン：

Influence of Fitness Quantization Noise on the Performance of Interactive PSO

Yu NAKANO Hideyuki TAKAGI

Abstract— **We analyze the influence of quantization noise in fitness values on the search performance of Particle Swarm Optimization (PSO) and propose methods for reducing the negative influence of the noise to help realize a practical Interactive PSO. First, we compare the convergences of PSO and genetic algorithms (GA) with several different levels of quantized fitness values and show that PSO has a higher sensitivity to quantization noise than GA. Second, we analyze the sensitivity of each of the three components that determine the subsequent generation's PSO velocities and show that the sensitivities of the three components are almost equivalent. This implies that we need to develop methods for reducing the effect of quantization noise on all three components of the PSO velocity. As one of the solution, we propose a method using the average location of multiple global bests of same fitness value and another method for multimodal searching spaces using sub-global bests obtained by clustering.**

I. Introduction

Interactive Evolutionary Computation (IEC) has been applied in a wide variety of fields such as the arts, engineering and others [7]. The IEC user plays the role filled by the fitness function in normal EC; we can say that an IEC is an EC where the fitness function is replaced with a human. A user can embed his or her preferences, intuition, knowledge, experiences, or *KANSEI* in general, through designing/optimizing target systems by giving his or her subjective evaluations to the EC-generated individuals. With the exception of this human evaluation stage, the procedures in IEC are otherwise identical to normal EC.

However, there are still problems impeding its practical application, with the problem of IEC user fatigue caused by interacting with a tireless computer being the most significant. Several approaches for solving this problem have been proposed [7]: learning the user's evaluation characteristics so that they can be modeled, improving the IEC interface used for displaying phenotypes of individuals and receiving the user-input IEC fitness values, accelerating the EC search, letting the IEC user intervene in the EC search, preparing other user models for predicting the IEC user's fitness, and others [2].

In this paper, we try to solve the user fatigue problem by extending the IEC framework and handling the EC portion of the IEC in a manner completely different from these previous approaches. The reason why IEC adopts EC is that the EC can search for the global optimum without requiring information about the searching space such as its gradient. Any optimization technique that has this characteristic can be used instead of EC in IEC. Particle Swarm Optimization (PSO) [4] is one such optimization technique, and using it we can create Interactive PSO (IPSO).

A comparison of genetic algorithms (GA) [3] and PSO suggests that IPSO may be a promising framework. It is said that PSO is more effective for searching the landscape of simple shapes, while GA is better than PSO for complex multimodal searching spaces, and we have confirmed this tendency using benchmark functions [6]. This tendency is theoretically reasonable because the information used by a PSO search are the relative direction to the global best (g_{best}) and the local best (l_{best}), and the precision of these directions is high when the searching landscape is simple.

Most IEC applications involve tasks where IEC users can reach the global optimum area within several generations. As IEC users cannot distinguish between similar individuals or do not give dramatically different fitness values for similar individuals, we can assume that the landscapes of the search spaces in most IEC applications are not complex. If this assumption is correct, we may be able to expect better searching convergence by replacing the EC of IEC with PSO.

However, an experimental comparison of Interactive GA (IGA) and IPSO using a simulator with a modeled pseudo-human that outputs relative fitness from a five point scale matched to benchmark functions did not meet this expectation. The performances of the IPSO are almost all equal to or lower than those achieved with IGA regardless of the complexity of the functions [6].

The objectives of this paper are (1) to analyze, with regards to quantization noise, why IPSO is not superior to IGA in tasks for which PSO is better than GA and (2) propose solutions to reduce the effect of fitness quantization noise. As one of the solutions, we propose an improvement to the calculation method for the components of the PSO velocity in order to make IPSO practical.

A simulation method for IPSO and IGA, along with a comparison of the results using the two procedures, is presented in section 2. We discuss the influence of quantization noise on the quantized fitness levels of IPSO in section 3 and analyze the noise sensitivities of each of the components that determine PSO velocity in section 4. Finally, discuss how to reduce the influence of the quantization noise on the IPSO performance and evaluate its efficiency in section 5.

Yu NAKANO was with Graduate School of Design, Kyushu University and conducted this work in those days.
Hideyuki TAKAGI is with Faculty of Design, Kyushu University, 4-9-1, Shiobaru, Minami-ku, Fukuoka, 815-8540 JAPAN, http://www.design.kyushu-u.ac.jp/~takagi/, TEL&FAX: +81-92-553-4555.

II. GA vs. PSO and Interactive PSO vs. Interactive GA

A. GA vs. PSO

It is said that PSO for less complex functions is faster than GA, and that GA for complex functions is faster than PSO. In a preliminary experiment, we compare them to clarify their different characteristics.

In this experiment, we use eight benchmark functions to compare the performance of PSO and GA. The optimization functions tasked are DeJong's F1–F5 functions [1] and three other functions (F6–F8).

F1, a quadratic function: $f(x_i) = \sum_{i=1}^{3} x_i^2$

F2, the Rosenbrock function:

$$f(x_i) = 100({x_1}^2 - x_2)^2 + (1 - x_1)^2$$

F3, a step function: $f(x_i) = \sum_{i=1}^{5} \lfloor x_i \rfloor$

F4, a 4-D function with standard Gaussian noise:

$$f(x_i) = \sum_{i=1}^{30} (ix_i^4) + Gauss(0, 1)$$

F5, the Shekel's foxholes function:

$$f(x_i) = \left[0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right]^{-1},$$

where 50 values of a_{ij} previously prepared.

F6, the Rastrigin function:

$$f(x_i) = 20 + \sum_{i=1}^{2} ({x_i}^2 - 10 \cos(2\pi x_i))$$

F7, the Griewangk function:

$$f(x_i) = 1 + \frac{1}{4000} \sum_{i=1}^{2} (x_i - 100)^2 - \prod_{i=1}^{2} \cos(\frac{x_i - 100}{\sqrt{i}})$$

F8, the Schwefel function: $f(x_i) = \sum_{i=1}^{2} (-x_i \sin(\sqrt{x_i}))$.

PSO and GA were run 50 times each, per function, under the experimental conditions list in Table I.

The average fitness values of the 50 bests at every 10th generation are shown in the Table II. These results match our expectations; they show that PSO for less complex functions converges faster than GA, and GA for more complex functions converges faster and found better solutions than PSO. In most cases, it seems natural that an IEC user will not change his/her evaluations dramatically when optimization parameters change slightly. This suggests that the psychological evaluation characteristics of an IEC user are not complex. This intuition is supported by the fact that IEC users obtain satisfactory solutions with a much smaller population size and within significantly fewer generations than in normal EC tasks. If this guess is true, we can expect that IPSO is faster than IGA.

B. Interactive PSO vs. Interactive GA

Unlike normal EC based on a fitness function, we need to add two fitness converters to normal EC in order to simulate IPSO; one simulates the discretization of relative fitness levels due to human evaluation, and the other converts from relative to absolute fitness levels. The first converter is also necessary to simulate IGA. We compare IPSO to IGA using these IPSO and IGA simulators with the aforementioned benchmark functions. As previously stated, an IEC user plays the role of the fitness function in EC which assigns a fitness value to each individual. We can therefore simulate the IEC user by using the benchmark functions as the fitness functions. Figs. 1 and 2 show the frameworks used in EC and IEC and an IEC simulation.

The first converter transforms absolute values obtained from a fitness function to relative discrete fitness values, i.e. it simulates human evaluation. An IEC user evaluates all individuals according to a scale with n rating levels, e.g. 1, 2, 3, 4, and 5 points, for each generation, while normal EC uses a continuous fitness function. We use five levels for the fitness values in our experiments. The human relative discrete fitness values on a 1 to 5 scale is simulated for the IPSO and IGA trials by dividing into five equal partitions the range between the highest fitness and the lowest fitness amongst all the individuals in a generation. The same discrete fitness value is assigned to an individual regardless of where it falls within a given partition, and the difference between the real fitness and the discrete fitness for each individual is the quantization noise in its fitness level.

The second converter converts from the relative fitness

TABLE I

Experimental conditions for PSO vs. GA

	PSO	GA
evaluation levels	5	
♯ of individuals	200	
max. generation	50	
constriction coefficient	0.9	-
max. velocity	1.0	-
crossover rate	-	0.7
crossover	-	two-point crossover
elitist rate	-	0.05
mutation rate	-	0.01
selection	-	roulette wheel selection

TABLE II

Experimental results for PSO vs. GA: average of 50 runs. Gray cells indicate which has better performance, PSO or GA.

gen.	func.	GA	PSO	func.	GA	PSO
1	F1	1.03	1.05	F5	10.8	12.2
10		0.01	0.00		0.99	1.96
20		0.00	0.00		0.99	1.19
30		0.00	0.00		0.99	1.19
40		0.00	0.00		0.99	1.19
50		0.00	0.00		0.99	1.19
1	F2	0.28	0.26	F6	3.42	4.49
10		0.01	0.00		0.34	0.57
20		0.01	0.00		0.12	0.01
30		0.01	0.00		0.08	0.00
40		0.00	0.00		0.06	0.00
50		0.00	0.00		0.05	0.00
1	F3	10.36	4.14	F7	0.02	0.02
10		2.70	0.00		0.00	0.00
20		0.38	0.00		0.00	0.00
30		0.16	0.00		0.00	0.00
40		0.08	0.00		0.00	0.00
50		0.04	0.00		0.00	0.06
1	F4	88.8	91.8	F8	-714.4	-697.8
10		21.26	4.28		-837.39	-830.1
20		6.01	0.83		-837.96	-833.3
30		1.20	0.41		-837.96	-833.3
40		-0.02	0.34		-837.96	-833.3
50		-0.65	0.23		-837.96	-833.3

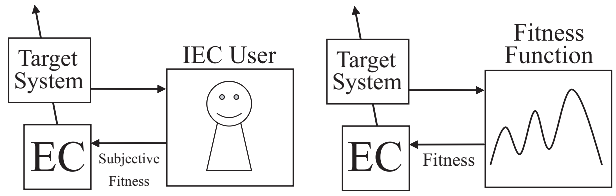


Fig. 1. Framework of IEC (left) and EC (right).

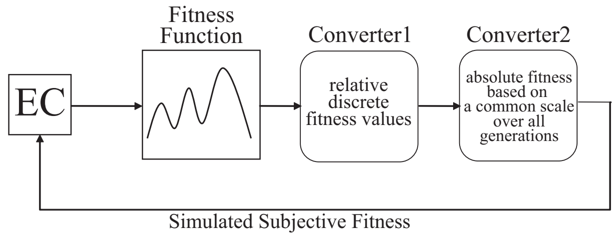


Fig. 2. Framework of IEC simulation

of a real or simulated IEC user to absolute fitness based on a scale which is common to all generations. The PSO offspring searching points are calculated using three velocity components, consisting of: current velocity; velocity towards g_{best}, that is the best searching point amongst all particles (or individuals) over all past generations; and velocity to the l_{best}, that is the best searching point for the current particle over all past generations. Since an IPSO user evaluates individuals with relative fitness, we cannot directly use fitness values from the past and cannot calculate g_{best} and l_{best} from relative fitness values. The second converter solves this problem.

The method for converting from relative fitness to absolute fitness is to copy some individuals from the n-th generation to the next $(n+1)$-th generation and correct all fitnesses using the average difference between the fitness values for these individuals in the n-th and the $(n+1)$-th generations [8]. Suppose an individual is rated as 3 points in the n-th generation, and the evaluation of the same individual is rated as 2 points in the $(n+1)$-th generation; we assume that the fitness values of all individuals increased 1 point on average versus those in the n-th generation, and thus add 1 point to all individuals in the $(n+1)$-th generation to obtain the absolute fitness. When we copy multiple individuals to the next generation, the average of difference in fitness is used to correct for relative fitness.

The average fitness values of the 50 bests at every 10th generation are shown in the Table III. Experimental conditions are the same as those listed in Table I. IPSO demonstrated poorer convergence than IGA for most cases. As the comparisons of PSO and GA showed that PSO was superior to GA for less complex functions, such as unimodal functions, and this seemed reasonable because PSO's velocity has a similar effect to that of a gradient, we expected that IPSO would be superior to IGA for the simple benchmark functions here. However, as can be seen from Tables I and III, reality did not match our expectations.

TABLE III

Experimental results of IPSO vs. IGA: average of 50 runs. Gray cells indicate better perfomance.

gen.	func.	IGA	IPSO	func.	IGA	IPSO
1	F1	6.17	5.67	F5	230.87	225.44
10		1.35	2.74		31.37	26.9
20		0.70	1.11		10.82	22.3
30		0.37	0.46		5.44	22.5
40		0.25	0.32		4.22	21.4
50		0.16	0.18		3.54	20.8
1	F2	126.0	93.8	F6	13.3	13.4
10		4.51	25.6		4.87	5.87
20		1.47	18.4		3.76	4.31
30		0.63	6.18		3.37	3.47
40		0.36	4.16		3.20	3.13
50		0.22	2.01		3.12	2.66
1	F3	15.8	16.4	F7	0.20	0.19
10		7.44	0.76		0.07	0.14
20		5.50	0.30		0.05	0.13
30		4.38	0.22		0.03	0.10
40		3.44	0.00		0.02	0.08
50		3.02	0.00		0.02	0.06
1	F4	141.3	139.6	F8	-494.9	-444.1
10		61.6	67.0		-720.4	-667.1
20		34.5	39.6		-739.8	-699.8
30		24.5	28.9		-750.0	-707.9
40		19.7	23.7		-757.8	-722.7
50		16.0	21.7		-760.7	-729.4

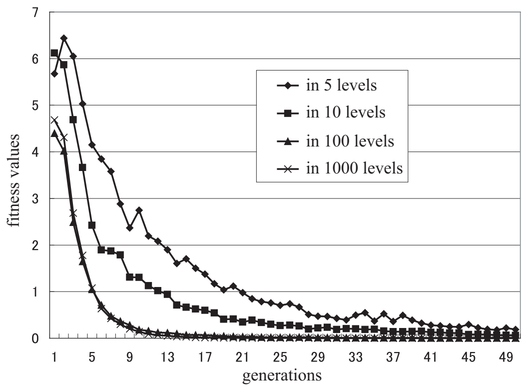
III. Tolerance of IPSO to Quantization Noise

The experiments in the previous section demonstrated that the convergence characteristics of IPSO vs. IGA for functions of different complexities did not match to those of PSO vs. GA. There are two possible explanations: the degradation is primarily caused by the first or second converter introduced to simulate IPSO (see Fig. 2). In this paper, we focus on the first converter, which is used to generate relative discrete fitness values, and analyze the IPSO characteristics.

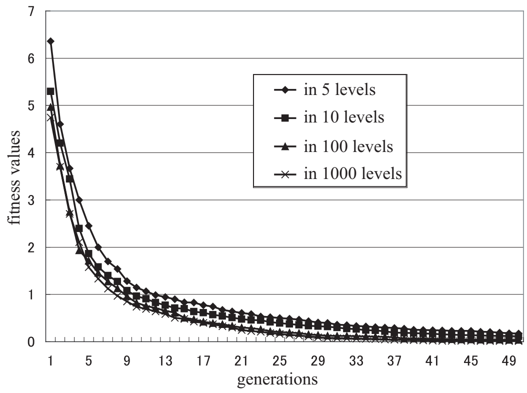
Our hypothesis is that IPSO is too sensitive to the quantization noise that is caused by the difference between actual fitness and the quantized fitness given by the IPSO user as discussed in the previous section. This noise is the main reason for the poor convergence. If this hypothesis is correct and we can develop measures to counteract the quantization noise in fitness, we may obtain similar performance in a comparison between IPSO and IGA as we did in the comparison between PSO and GA. Since it is expected that the search space landscapes of IEC tasks are simple, we hope to achieve better performance with the improved IPSO.

To test our hypothesis, we observed IPSO performance along with quantization noise as we changed the number of quantization levels used for fitness (for 5, 10, 100, and 1000 levels) at the first converter mentioned in the previous section. Experimental conditions were otherwise the same as listed in Table I. Figs. 3 and 4 show these experimental results for two of eight benchmark functions.

The new experimental results demonstrated that the convergence performance of IGA did not greatly depend on the number of quantization levels in fitness. On the other hand, the convergence performance of IPSO became better as fitness quantization levels increased, i.e in inverse proportion

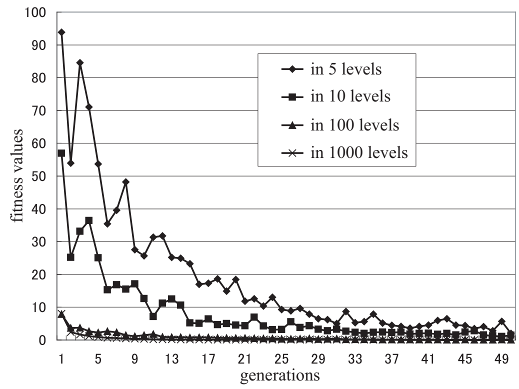


(a) IPSO search for the F_1 function.

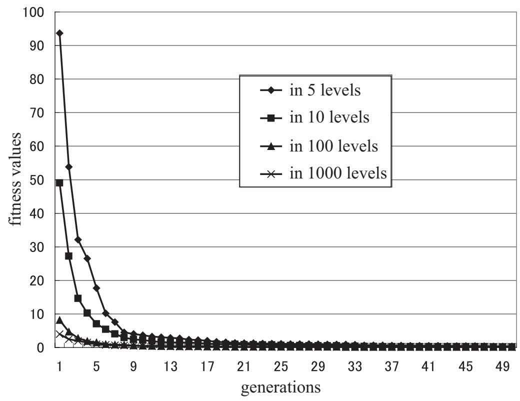


(b) IGA search for the F_1 function.

Fig. 3. IPSO and IGA searches for the F_1 function with four different quantization levels.



(a) IPSO search for the F_2 function.



(b) IGA search for the F_2 function.

Fig. 4. IPSO and IGA searches for the F_2 functions with four different quantization levels.

to the quantization noise.

These experimental results clearly show that IPSO performance is influenced by the amount of quantization noise. The same results were also obtained for the other six benchmark functions. We therefore conclude that our hypothesis is correct, and unlike IGA (and GA), IPSO (and PSO) suffers from excessive sensitivity to quantization noise.

IV. Analyzing the Tolerance of Each Component of IPSO Velocity to Quantization Noise

What component of IPSO is excessively sensitive to fitness quantization noise? If it is possible to find a single component which is responsible for the greatest sensitivity, we can focus our efforts on improving it. All individuals share a common g_{best}, and each individual remembers its l_{best}. The g_{best} is the best searching point in the searching histories of all individuals, while the l_{best} is the best searching point in the searching history of a given individual. There is only one g_{best}, while each individual has a l_{best}.

A PSO's velocity is determined by the velocity components of g_{best} and l_{best} and a momentum velocity. A term directed towards the g_{best} determines the searching direction of the whole PSO population, while a term directed towards the l_{best} tries to search for the best searching point in the individual's past.

In this section, we evaluate the noise tolerance of each component determining PSO velocity. The velocity is expressed according to the following equation.

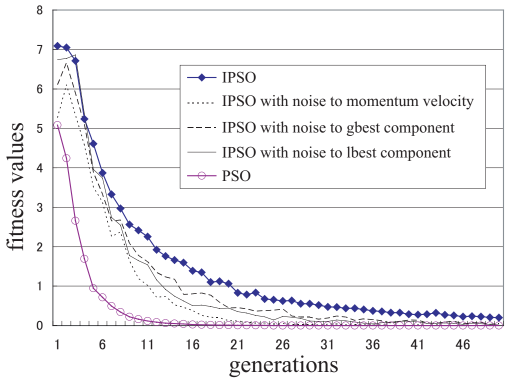
$$\overrightarrow{V}_{new} = w_1\overrightarrow{V}_{old} + w_2(\overrightarrow{P}_{g_{best}} - \overrightarrow{x}_i) + w_3(\overrightarrow{P}_{l_{best}} - \overrightarrow{x}_i)$$

V: velocity,
w: weight coefficient of each term,
P: positions of g_{best} and l_{best},
x: current position of each individual

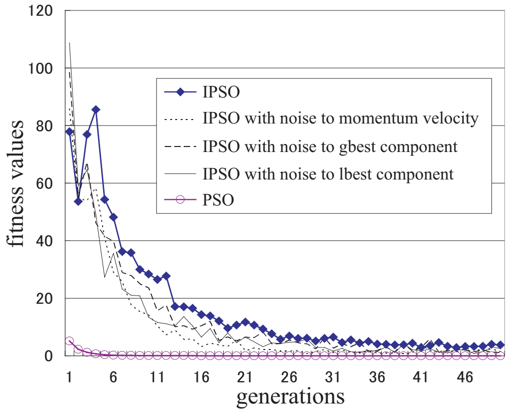
To evaluate the effect of quantization noise on each of the three velocity components, we discretize the fitness of one component while using unquantized fitness values for the remaining two components. We test three combinations of discrete fitness (with quantization noise) and noiseless continuous fitness. The experimental conditions used are the same as those from the previous section.

Fig. 5 shows the experimental results for the benchmark functions of F_1, F_2, and F_5. The results of IPSO from the previous section and normal PSO are also illustrated in these figures for reference.

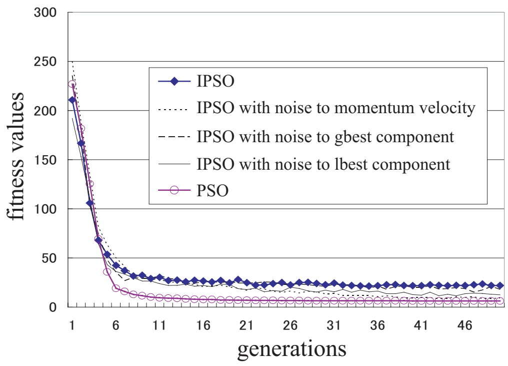
The experimental results show that (1) IPSO performance becomes worse than normal PSO when any of velocity components include quantization noise in fitness and (2) there is no signficant difference in sensitivity to quantization



(a) IPSO search with the F_1 function.



(b) IPSO search with the F_2 function.



(c) IPSO search with the F_5 function.

Fig. 5. IPSO search when only one of three velocity components includes quantization noise.

noise among the three velocity components. In conclusion, we cannot improve the IPSO's performance by improving the noise sensitive of just one or two components, but rather must solve the effect of noise sensitivity for all components.

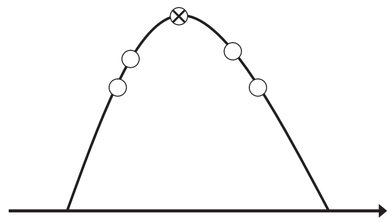
V. Improving the Noise Tolerance of the g_{best} Velocity

A. Two Proposals

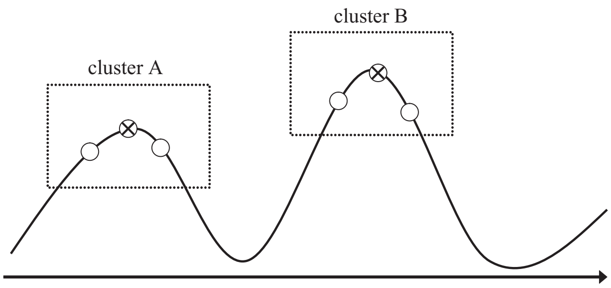
We introduce a method for reducing the negative influence on the search performance of all individuals caused by the fitness value quantization noise in the calculation of g_{best}. To accomplish this, we use the location of multiple g_{best}s that are assigned the same best fitness value using one of the two below listed approaches.

(a) Let g_{best} = the center of gravity over all g^i_{best}s

In this method, g_{best} is calculated as the center of gravity over the multiple g_{best} candidates to which the same highest fitness is given (see Fig. 6(a)). Because fitness values are taken over some fitness range as described for the converter of section II-B, it is inevitable that the same discrete value will be given to individuals which fall in the same partition. Consequently, it is common for multiple individuals to have the same highest fitness value. If the g_{best} is chosen at random from the multiple g^i_{best}s, the actual best may not be chosen, due to the quantization of fitness levels. This method (a) minimizes the fluctuation of the quantization noise and stabilizes the g_{best}. When the multiple best g^i_{best}s are near to the global optimum, the method may accelerate convergence as in the case of Fig. 6(a).



(a) method (a).



(b) method (b).

Fig. 6. Determining the g_{best} using methods (a) and (b). $\otimes$ and $\bigcirc$ are g_{best} andg^i_{best}'s, respectively.

(b) Let g_{best} = the center of gravity for the g^i_{best}s of each cluster

In this method, the g_{best} candidates are grouped into clusters, and g_{best} for each cluster is calculated as being at the center of gravity of themultiple $gbest$s in its cluster. When an IPSO task is a multimodal function, and the fitness values of local optima are near to that of the global optimum, the multiple g_{best} candidates may be spread over a very wide searching space. We cannot expect that their center of gravity is close to the location of the actual g_{best} in this case, and the method outlined in (a) may actually reduce IPSO performance. Method (b) solves this problem by introducing clustering.

In method (b), local optima are separated by clustering g_{best} candidates. Let's call the center of gravity of the multiple g^i_{best} candidates in each cluster sub-g_{best}. Each individual uses its nearest sub-g_{best} instead of the g_{best} as shown in Fig. 6(b). As method (b) provides sub-g_{best}s that are close to the local optima, it improves upon method (a) when operating in a multimodal searching space. In the next section, we discuss how the *K*-means method [5] can be used to determine the clusters.

B. Evaluations of the two Proposed Methods for Determining a More Accurate g_{best}

The methods (a) and (b) from above are applied to the eight benchmark functions used in section II with the experimental conditions listed in Table I. Since the number of g_{best} candidates produced by a population with 20 individuals is few, we set 2 as the number of clusters for the *K*-means method.

Figs. 8(a) – (c) show the average convergence of the best fitness after 50 runs of 3 of the 8 benchmark functions.

The method (a) worked well for the unimodal functions of F_1 and F_2, while method (b) did not work for any benchmark functions. F_5 includes many local minima and looks quite different from the evaluation characteristics of IPSO users, so this may explain the poor result.

Next, both methods (a) and (b) were applied to the 2-D Gaussian mixture model shown in Fig. 7. This function looks closer to a human evaluation characteristic than does the tricky function of F_5. The 2-D Gaussian mixture model is expressed as $F = -2.8 * N(-2, 1.5) - 3.4 * N(3.4, 1) - 2.1 * N(1, 1) - 1.8 * N(-4, 0.8)$, where $N(\mu, \sigma)$ is a 2-D normalized Gaussian function and μ and σ of x-axis and y-axis are the same. The results in Fig. 8(d) show that there is no significant difference until after around 50 generations, whereafter method (b) outperforms the others. Although the number of clusters that method (b) used in this experiment was only two, and the Gaussian mixture model has four local minima, the clustering of method (b) seemed to work well and reduced the estimation error of g_{best} near the global optimum.

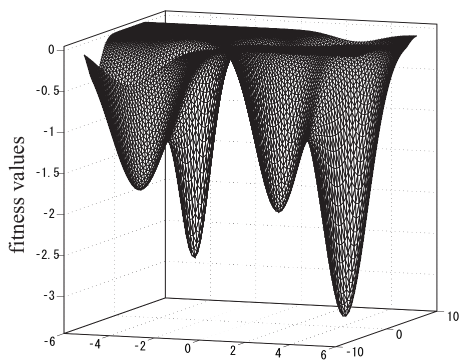


Fig. 7. The Gaussian mixture model used for an experimental evaluation.

VI. Discussion and Conclusions

We have made it clear that (1) the convergence of PSO is deeply influenced by noise in fitness, and the noise sensitivity of PSO is higher than that of GA and (2) IEC cannot avoid quantization noise in fitness levels and due to this noise sensitivity, IPSO performance becomes worse than that of IGA. Following from these observations, we analyzed which factor in IPSO is most susceptible to noise and determined that (a) the velocity components which determine the searching points for the next generation are the main points influenced by the noise, and (b) the influence of the noise is similar on all three velocity components. (c) Consequently, we must find solutions for reducing the noise sensitivity for each of the three components.

We proposed two methods for reducing the influence of the fitness quantization noise on the determination of g_{best}. From the relationships between the types of IEC tasks and the comparison of PSO and GA, we can expect that IPSO performance should exceed that of GA if we can reduce the influence of the noise. The application of the methods to the determination of l_{best} is not described in this paper; applying the same idea of using the centers of gravity gravity of l_{best} candidates as the new l_{best}s is the next step of our research.

The following phase will be to analyze the effect of the conversion in IPSO of fitness levels from relative to absolute values and to clarify which factors are reducing IPSO performance. Based on that analysis, we will develop methods to counteract these factors and make IPSO practical.

References

[1] K. De Jong, "An analysis of the behaviour of a class of genetic adaptive systems," *PhD thesis*, University of Michigan (1975).

[2] S. Henmi, S. Iwashita, and H. Takagi, "Interactive Evolutionary Computation with Evaluation Characteristics of Multi-IEC Users," *2006 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC2006)*, Taipei, Taiwan, pp.3475–3480 (Oct., 2006).

[3] J. H., Holland, "Adaptation in Natural and Artificial Systems," *University of Michigan Press* (1975).

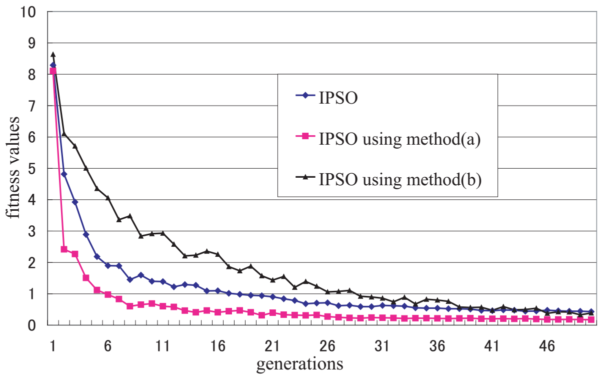
[4] J. Kennedy and R. Eberhert, "Particle Swarm Optimization," *IEEE Int. Conf. on Neural Networks*, Piscataway, NJ, pp.1942–1948 (1995).

[5] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of Fifth Berkeley Symp. on Math. Statist. and Prob.*, Vol. 1, pp.281–297 (1967).

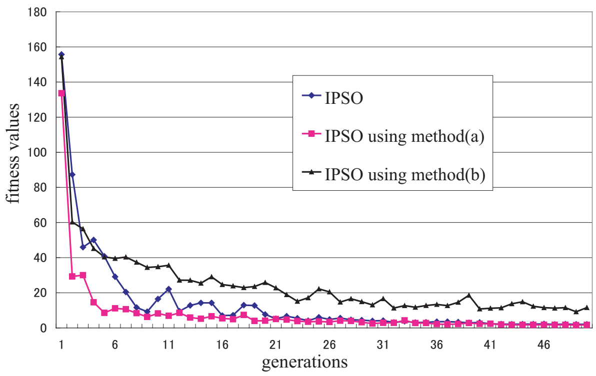
[6] Y. Nakano and H. Takagi, "Influence of Quantization Noise in Fitness Values to the Performance of Interactive PSO," *Intelligent System Symposium (FAN Symposium)*, Nagoya, Japan, pp.325–328 (Aug., 2007) (in Japanese).

[7] H. Takagi, "Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp.1275–1296 (2001).

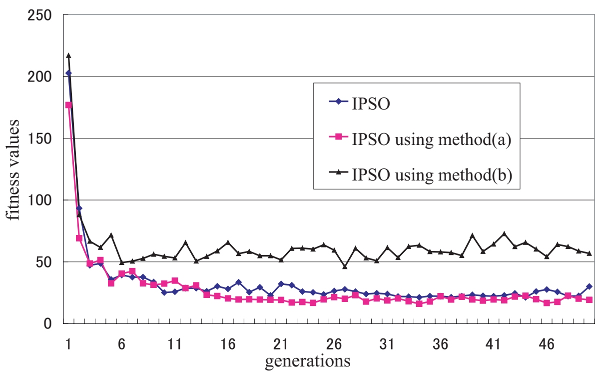
[8] S. Wang and H. Takagi, "Improving the Performance of Predicting Users' Subjective Evaluation Characteristics to Reduce Their Fatigue in IEC," *J. of Physiological Anthropology and Applied Human Science*, vol. 24, no. 1, pp.81–85 (2005).



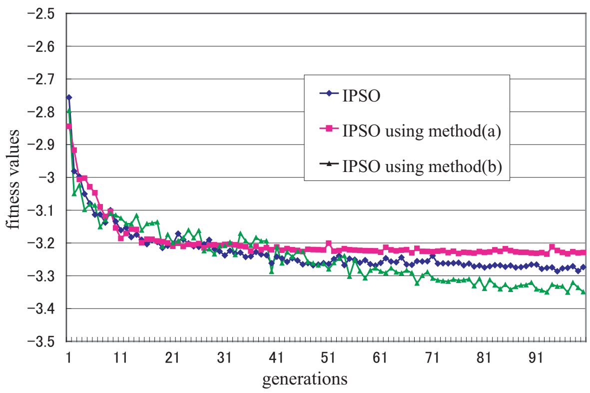
(a) IPSO search with the F_1 function.



(b) IPSO search with the F_2 function.



(c) IPSO search with the F_5 function.



(d) IPSO search with the Gaussian mixture model in Fig.7.

Fig. 8. IPSO search with the four functions using the method(a) and method(b).