Analysis of Noisy Time-series Signals with GA Involving Viral Infection with Tropism

Yuji Sato
Hosei University
3-7-2 Kajino-cho Koganei-shi
Tokyo 184-8584 JAPAN
+81-42-387-4533
yuji@k.hosei.ac.jp

Yuta Yasuda
Hosei University
3-7-2 Kajino-cho Koganei-shi
Tokyo 184-8584 JAPAN
+81-42-387-4533
i05t0010@k.hosei.ac.jp

Ryuji Goto
Hosei University
3-7-2 Kajino-cho Koganei-shi
Tokyo 184-8584 JAPAN
+81-42-387-4533
i06t9001@k.hosei.ac.jp

ABSTRACT
In this paper we report on a study in which genetic algorithms are applied to the analysis of noisy time-series signals, which is related to the problem of analyzing the motion characteristics of moving bodies (distance, bearing, course, velocity, etc.) by covertly sampling the sound of moving objects with submarine monitoring systems that track moving objects travelling on or through the water. In particular, we propose improving the system's ability to search through noisy data by grafting viruses onto the chromosomes used in genetic algorithms. Specifically, we propose a search method that can cope robustly with noise through the cooperative action of a wide-area search implemented by host chromosomes and a local search implemented by viruses grafted onto these chromosomes. To improve the infection rate, we also impose limits on the types of host entity that can be infected by viruses. By conducting evaluation tests in computer simulations, we show that the proposed technique can achieve a better rate of convergence and is capable of searching for a solution with fewer entities.

Categories and Subject Descriptors
J.0 [**Computer Applications**]: General

General Terms
Design, Experimentation, Performance, Verification.

Keywords
Genetic Algorithms, Virus Infection, Virus Evolution, Tropism, Time-series Problem, Noisy Optimization, Inverse Problem.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

1. INTRODUCTION
Genetic algorithms (GA) can be used to solve various classes of problems in game software and software designed to solve real-world problems. For example, they have been applied to multi-purpose optimization [1–3], time-series prediction [4, 5], the optimal analysis of noisy data [6–9], and the analysis of implicit functions [10]. As an example of an application that involves addressing each of these classes of problem simultaneously, we have been researching an underwater tracking device that tracks an unidentified object moving on or below the ocean surface. A device of this sort works by measuring the sound produced by the moving object (which we will refer to as "sound", although it may also include noise from other sources such as turbulence), and analyzes this audio signal to obtain information such as the location of the moving object, its motion vector, and its acoustic fingerprint (obtained by processing the audio spectrum to eliminate Doppler effects).

We have already published the results of several studies on the feasibility of using GA to implement such a device [11–14]. In practical situations, when an underwater observer has detected an unknown object moving in or on the water, it must be able to efficiently analyze the moving object's motion and home in on an advantageous position with regard to the object without revealing its own presence to the object. This means the observer has to perform passive motion analysis by measuring the time-series variation of the arrival angle (bearing) of the moving object's sound with a directional listening instrument, since the use of active methods would involve emitting acoustic or radio waves from the observer. If the object's bearing can be measured accurately, then its motion can be precisely analyzed and a solution can be obtained in a short time.

However, this approach has three drawbacks. First, a directional listening instrument large enough to make bearing measurements with sufficient accuracy for motion analysis would have to be quite cumbersome. Second, to reduce the adverse effects of bearing measurement errors on the accuracy of motion analysis, the observer would have to perform circuitous movements to increase the temporal variation of the object's bearing. And third, in optimization analysis methods that are based on ordinary multiple simultaneous equations, the observer has to incorporate linear motion at non-constant velocity into the analysis period in order to obtain a solution. The second and third drawbacks make efficient approximation difficult to achieve. It is thus difficult to derive a realistic solution by analytical methods, so it is hoped

that stochastic search methods such as GA will be more suited to the task.

On a related issue, we have already shown that it is possible to analyze the position of an audio source by applying GA to the analysis of noisy continuous time signals [13]. In this paper we report on the results obtained when attempting to improve the search capabilities with respect to noisy data by grafting viruses on to part of the chromosomes used for GA.

In section 2 we formulate the problem to be solved and present an overview of the GA techniques used in our previous report [13]. In section 3 we propose a GA technique that involves viral infection and a technique for using tropism to improve the infection rate. In section 4 we describe the simulations we performed to evaluate this technique, and in the final sections we discuss our results and finish with a conclusion.

2. ANALYSIS OF TIME-SERIES SIGNALS WITH GENETIC ALGORITHM

2.1 Formulations of Time Series Signals

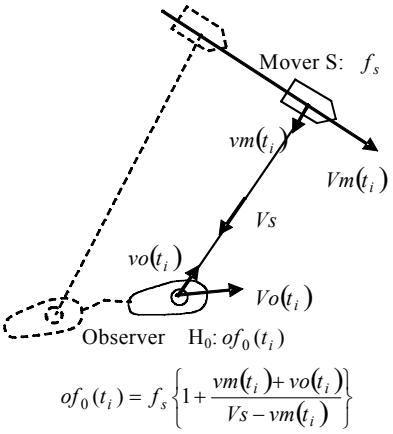

Figure 1. A sound path from the mover to the observer.

During the mover navigates in constant velocity, sound source frequency radiated by propulsion system is periodic and constant frequency. However observed frequency at hydrophone on the observer has Doppler effects. Figure 1 shows a sound path from the mover to the observer. Equation in Fig. 1 shows relationship between sound source frequency and observed frequency. This equation is based on Doppler Effect Theory. *fs* of the equation is sound source frequency. $f_{p0}(t_i)$ is observed frequency at hydrophone H_0. $Vm(t_i)$ is mover's approaching velocity along sound path at time t_i, $Vo(t_i)$ is observer's approaching velocity along sound path. They have positive polarity while approaching, negative while receding. V_s is propagation velocity of sound in sea.

Figure 2 shows an example of the audio signal $S(t)$ observed by hydrophone H_0. When the period of this signal is T and angle velocity is ω_0, the result of Fourier Series Development is expressed in Equation (1).

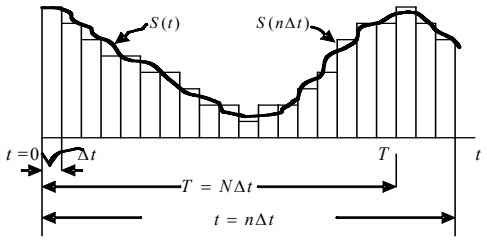

Figure 2. Quantum expression of $S(n\Delta t)$ for observed sound signal $S(t)$.

$$\left.\begin{aligned} S(t) &= \sum_{k=-\infty}^{\infty} C_k e^{jk\omega_0 t} \\ &\text{where} \\ C_k &= \frac{1}{T}\int_0^T S(t)e^{-jk\omega_0 t}\,dt \end{aligned}\right\} \quad (1)$$

Equation (1) contains integral equations, and is not suitable for computer processing. Consequently, it is necessary to transform Equation (1) into Equation (2) expressed in quantum.

$$\left.\begin{aligned} S(t) &= S(n\Delta t) \\ &= A_0 + A_1 \sin\left(2\pi f_p n\Delta t + \phi_1\right) \\ &\quad + A_2 \sin\left(4\pi f_p n\Delta t + \phi_2\right) \\ &\quad \vdots \\ &\quad + A_k \sin\left(2k\pi f_p n\Delta t + \phi_k\right) \\ &\text{where} \\ A_0 &= \Delta t f_p \sum_{n=0}^{1/\Delta t f_p} S(n\Delta t), \\ A_k &= \sqrt{a_k^2 + b_k^2}, \\ \phi_k &= \tan^{-1}\frac{a_k}{b_k} \end{aligned}\right\} \quad (2)$$

$S(t)$ can be transformed into $S(n\Delta t)$ after transformation by Euler's formula on the condition of $n \geq N$. Here, A_0 in (2) is the bias (direct current) component. A_1, A_2, A_3, …, A_k, … are amplitudes of harmonic frequencies, and a_k, b_k, φ_k express the k-th harmonic frequency components. f_p is the fundamental frequency. Accordingly, time series signal $S(n\Delta t)$ is defined in term of components in Equation (2). The problem addressed in this paper can be formulated as an inverse problem involving complex implicit function. Here is necessary to find the components of a time series periodic signal by counting backward from noisy time series sampled data of its $S(n\Delta t)$ obtained with measurement equipment.

2.2 Method of Applying Genetic Algorithms

2.2.1 Definition of Chromosomes

Figure 3 shows the constitution of a chromosome of the analysis GA. In analysis for characteristics of signals, low-pass filter eliminates higher frequency of signals, thereby higher frequencies are not as useful as lower frequencies and sometimes harmful for the analysis. Therefore, this research set 6 as maximum of k in Equation (2). Thereby components to analyze are frequency and Fourier coefficient up to 6 since A_0 is dependent upon f_p as shown in Equation (2). Consequently one chromosome consists of 13 sub- chromosomes f_p, a_1, b_1, a_2, b_2, a_3, b_3, a_4, b_4, a_5, b_5, a_6, b_6.

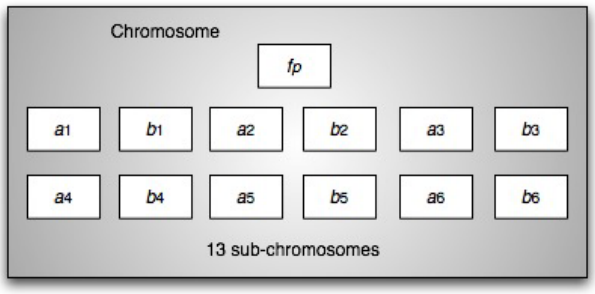

Figure 3. The constitution of chromosome of the GA. This chromosome consists of 13 sub-chromosomes.

2.2.2 Fitness Function

In cases where the fitness function is highly sensitive to differences between a genetic entity's own solution and the correct solution, the search range of the correct solution is narrow, and in cases where the sensitivity is low, the search range is wider. In the former, the range in which the extreme values of the solution can exist is limited, which is effective in cases where there are few deceptive solutions [15], while the latter is effective in cases where there are many deceptive solutions and it is easy to escape from a deceptive solution. A fitness function for acoustic analysis GA is shown in Equation (3).

$$fitness = \frac{CJ}{\sum_{n=0}^{J-1} \left| oS(n\Delta t) - esS(n\Delta t) \right|} \tag{3}$$

This fitness function produces a higher value when the absolute difference between the sampled engine noise signal amplitude $oS(n\Delta t)$ and the amplitude $esS(n\Delta t)$ of the audio signal estimated by the chromosome has a small average value. Since there are many deceptive solutions, the search range is broadened by reducing the sensitivity. C is a constant and J is the number of samples.

2.2.3 Genetic Manipulation

In generation shifts, elite individuals E first carry forward to next generation, the individuals which are selected in descending order of fitness from all individuals P. Next, parent individuals are selected from P by roulette selection. Genetic manipulation then generates child individuals from one pair of parent individuals. These child individuals make up the rest of individuals $P - E$ in next generation. In genetic operations, we employ one-point crossover as crossover, and mutation based on the rate set beforehand. On mutation rate, sub chromosome f_p starts to converge earlier than the other sub chromosomes, we reduce mutation rate for f_p from ones for the other sub chromosomes accordingly. All sub chromosomes start to converge around 10 generation, we reduce mutation rate for all sub chromosomes accordingly. Crossover and mutation are performed on each 13 sub-chromosome independently.

3. VIRAL INFECTION WITH TROPISM

3.1 Basic Concepts

We have already shown that it is possible to analyze the position of an audio source by applying GA to the analysis of noisy time-series data. On the other hand, when standard GA is applied to the analysis of noisy time-series data, we found that it requires more generations and entities to search for a solution than in the noise-free case. This is because local searching is not so easy when the data is assumed to be noisy. Specifically, the results tend to reflect the characteristic weakness of GA in local searches coupled with strength in wide-area searches. To tackle this problem, we propose here a technique that involves grafting viruses onto part of the chromosomes used in GA as a means of improving the local search capabilities of GA. This operation of virus infection is inspired by virus evolutionary theory [16]. Figure 4 shows the basic concept of genetic algorithms involving virus infection. Our aim is to implement a search technique that is robust against noise by encouraging cooperative interaction between a wide-area search based on host entities and a local search based on viruses grafted onto the chromosomes of these host entities.

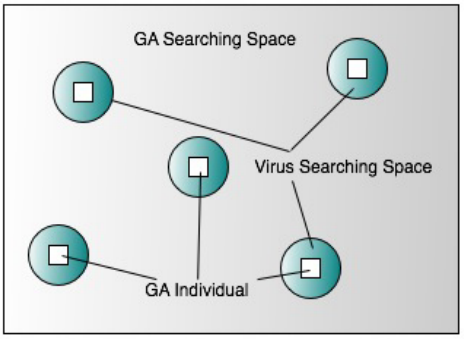

Figure 4. The basic concept of genetic algorithms involving virus infection.

3.2 Virus Infection Operation and Virus Evolution

Here, we define the virus infection operation and the virus evolution mechanism. First, we will define the virus entities and the virus infection operation. The chromosomes of the virus entities are defined as partial genetic information (schema) of a host chromosome. As mechanisms for the infection of hosts with viruses, we will consider reverse transcription operations and incorporation operations [17]. As shown in Figure 5, in a reverse transcription operation, the partial genetic information of the virus is copied into the host entity, and an incorporation operation, the virus is copied to part of the chromosome (partial genetic information) of the host entity. In an incorporation operation, part of the chromosome information held by the virus may be lost.

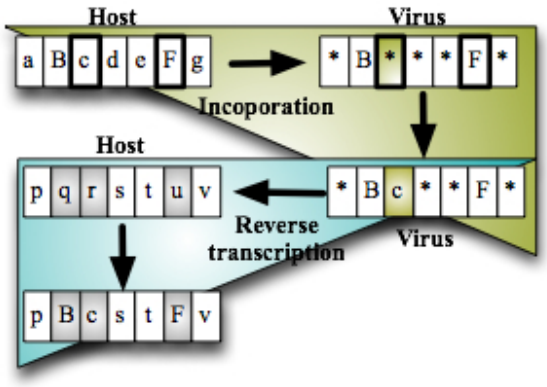

Figure 5. Virus infection operation.

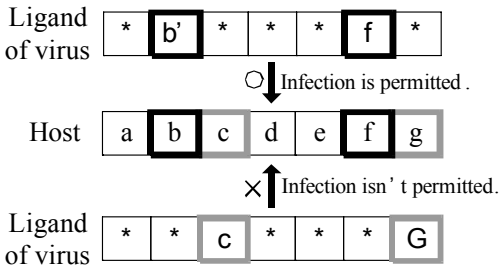

Figure 6. An example of how the possibility of infection is controlled by tropism.

Next we will consider the virus evolution. As an evaluation index for virus evolution, we define the evaluation value, infectiousness and vitality of the virus [17]. A virus's evaluation value *fitvalue* is defined as the improvement of the host due to reverse transcription. If S is a group of entities infected with virus i, *fithostj* is the evaluation value of host j before reverse transcription, and *fithostj* is the value after reverse transcription, then *fitvirusi* is defined as shown in Equation (4) below:

$$\left.\begin{aligned} & fitvirus_i = \sum_{j \in S} fitvirus_{i,j} \\ & where \\ & fitvirus_{i,j} = fithost_{j'} - fithost_j \end{aligned}\right\} \quad (4)$$

The infectiousness of a virus is defined so that the infectiousness increases as a virus becomes more successful at reverse transcription, as expressed by Equation (5) below:

$$\inf rate_{i,t+1} = \begin{cases} (1+\alpha) \times \inf rate_{i,t} \\ \quad if \;\; fitvirus_i > 0 \\ (1-\alpha) \times \inf rate_{i,t} \\ \quad otherwise \end{cases} \quad (5)$$

The vitality of a virus is defined by Equation (6) below. As viruses become more successful at reverse transcription, they retain more vitality.

$$life_{i,t+1} = r \times life_{i,t} + fitvirus_i \quad (6)$$

In equation (6), the variable r ($0 \le r \le 1$) is a real number. When the vitality of a virus reaches zero, a new virus entity is created in its place.

3.3 Proposal of Increased Infection Efficiency Using Tropism

In the abovementioned virus infection, all the entities used to perform GA are host candidates, and infection is performed stochastically on all the entities. Although the evaluation value, infectiousness and vitality of a virus may evolve to higher values, this does not mean that the evaluation value of a host entity infected by a virus with high infectiousness or the like will also increase in a similar manner. A highly infectious virus may sometimes have an adverse effect on its host, causing the host entity to become extinct. In this paper, we propose addressing this issue by employing tropism to improve the infection efficiency. In biology, the term tropism describes how viruses are only able to propagate to and infect a limited range of host species, or a limited range of tissues or organs at the individual level [18]. This behavior arises because host entities are liable to be destroyed and become extinct when chaotic viral propagation and infection occurs repeatedly, which is also disadvantageous for the viruses themselves.

One way in which tropism occurs in the natural world is through the existence of proteins called ligands around the outside of viruses, which can only infect cells that have receptors that fit these ligands. Using this natural mechanism as a model, we introduced a tropism mechanism into the genetic algorithm associated with viral infection by treating the partial genetic information of a host entity as a receptor and providing the viruses with identical information as a ligand.

Figure 6 shows an example of how the possibility of viral infection is controlled by tropism. In this figure, the values a through g of the gene locus of the host's chromosomes are assumed to be real numbers. Asterisks represent "don't care" states where any real value can be accepted. In the chromosomes of a virus entity, the information of the gene locus surrounded by a thick border indicates the ligand. The difference between b and b' in the gene locus values is assumed to be smaller than a preset threshold value, while the difference between g and G is assumed to be larger than the threshold value. In the example shown here, infection can take place when the difference between the ligand and host entity receptor is less than the preset threshold (e.g., b and b'), but is not allowed when the difference is larger than the threshold (e.g., g and G). A tropism mechanism is implemented by limiting viral infection to chromosomes used in GA based on threshold value information and on ligand information defined on the chromosomes of the virus entity. Infection is performed stochastically on host candidate entities that satisfy the infection conditions.

4. EVALUATION TESTS

4.1 Evaluation Method

Figure 7 shows an overview of the evaluation system, which consists of an observed value generator unit and a GA computation analysis unit. In the observed value generator unit, theoretical values of the time-series data are first generated for the

13 types of variables shown in section 2, and then the values obtained by adding normally-distributed errors to these theoretical values are stored in a database (DB) as observed values. In the GA computation analysis unit, the initial values of the estimated values are first randomly generated, and the estimated values are updated by repeatedly performing genetic operations while calculating the goodness-of-fit based on these estimated values and the observed values stored in the DB. We used this evaluation system to perform three evaluation experiments. First we conducted a comparative evaluation of the search capabilities of GA associated with the GA with viral grafting compared with standard GA as used in previous reports. We then investigated the robustness of the system with different numbers of host entities (either 1000 or 2000), with the other conditions left the same as in the first test. And in the final experiment, we investigated in greater detail how the search performance varied with different numbers of host entities (varied in increments of 100).

The parameters used in these evaluation tests are shown in Tables 1 and 2. Table 1 shows the observer parameters. The observer speed is assumed to be low because the engine noise generated by the observer must not be detectable by the moving object, and must not have an adverse effect on the sampling of the moving object's sound. We set 500Hz as fundamental frequency of the sound signal to be analyzed. The sampling frequency was set based on a Nyquist frequency, which is the frequency required for reconstruction of the audio signal (satisfying the sampling theory). Table 2 shows the GA analysis parameters. Infection rate means the rate that virus performs reverse transcription, then copy rate and cut rate means the rate that virus performs incorporation. If *fitvirus* > 0, virus incorporates partial codeword of a host individual on copy rate. While virus loses its codeword on cut rate, if *fitvirus* ≤ 0. Incorporation rate means the rate that each sub chromosome gets or lose codeword at incorporation or generating virus individuals.

Table 1. The parameters of observation

Fundamental Frequency of Sound (contain Doppler Effect)	500Hz
Observation Error, Noise	0.1 %Max
Sampling Frequency	6.0 kHz

Table 2. The GA analysis parameters

Population size	3000
Maximum generation	100
Crossover rate	0.8
Mutation rate for f_p	0.0001
generation ≥ 7	0.00005
Mutation rate for a_k, b_k	0.0005
generation ≥ 7	0.0001
Virus population size	Population size * 0.01
Initial infection rate	0.01
Maximum infection rate	0.1
Copy rate	0.2
Cut rate	0.15
Incorporation rate	0.1

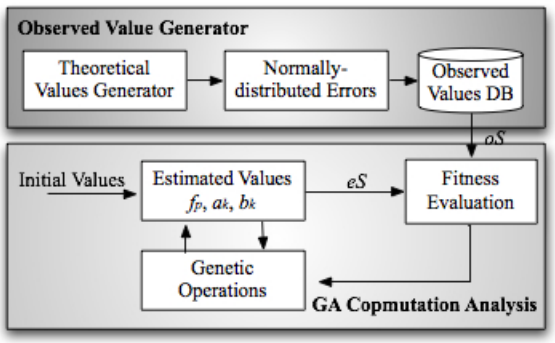

Figure 7. Overview of the evaluation system.

4.2 Evaluation Results

4.2.1 Performance comparison of GA with viral grafting and standard GA

On the theme of analyzing the characteristics of noisy time-series signals, Fig. 8 shows the relationship between the number of GA generations and the analysis errors. This figure compares the performance of three techniques: standard GA, GA with a viral infection function, and GA with a viral infection function that includes tropism. The results shown here were obtained by performing 100 tests under the same conditions with each method. Here, the analysis errors are the mean error of 12 of the parameters shown in section 2 (with the exclusion of frequency). In Fig. 8, it can be seen that no particular differences in search capabilities were observed in any technique up to the 20th generation. However, the search capabilities of standard GA tailed off after 20 generations, while there was no change in the search capabilities of GA with a viral infection function. The error at the 100th generation was about 10 times smaller in GA with a viral infection function compared with standard GA.

4.2.2 Survey of robustness when the number of host entities is changed

Figure 9 compares the results obtained from standard GA performed with 3000 entities and from GA with viral infection performed with 2000 or 1000 entities. This figure shows the results obtained from 100 trials in each case. The standard GA in this figure is the same as the standard GA in Fig. 8 when set with 3000 entities. From Fig. 9, it can be seen that when the number of hosts in GA with viral grafting is set to 1000, its search capabilities are higher than those of standard GA. When population size is 2000, tropism does not impact searching ability, as Figure 9. On the other hand, when population size is 1000, tropism increases the searching ability and reduces effect of population size on the searching ability.

4.2.3 Relationship between changes in the number of host entities and tropism effects

Figure 10 and 11shows the details of the relationship between the number of GA generations and the error magnitude when the number of host entities is varied in standard GA and GA with viral infection including tropism. In this figure, the number of host entities is varied from 500 to 2000 in increments of 100, and the results in each case indicate the performance in 100 trials.

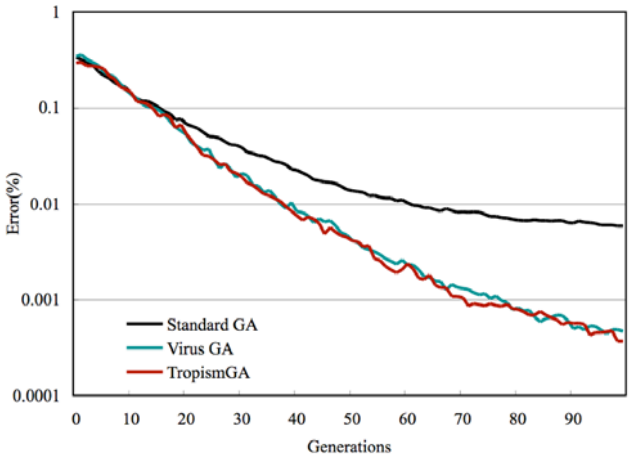

Figure 8. The relationship between the number of GA generation and the analysis errors.

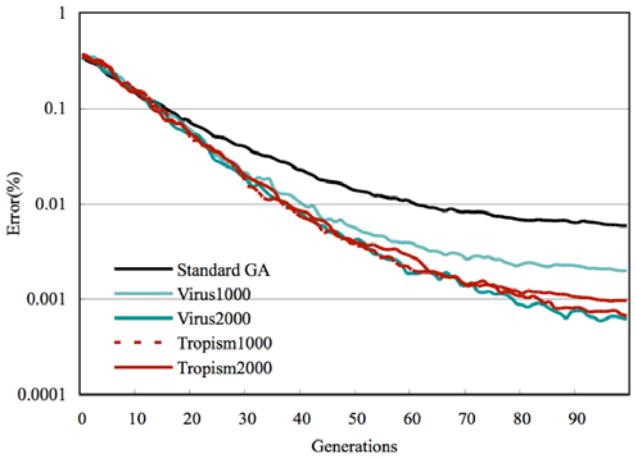

Figure 9. Comparison of search capabilities with different numbers of host entities.

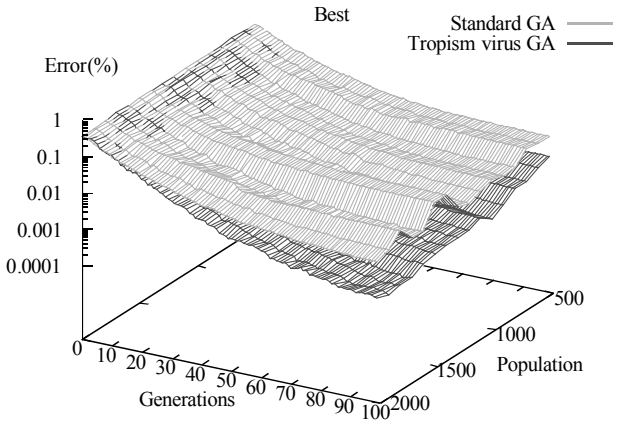

Figure 10. Error of the best individual when the number of host entities is varied in standard GA and GA with viral infection including tropism.

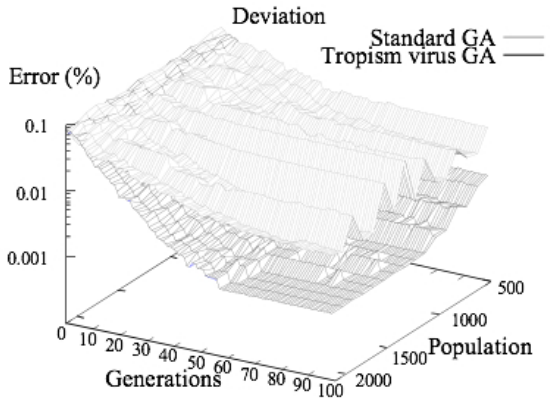

Figure 11. The deviation for error of the best individual when the number of host entities is varied in standard GA and GA with viral infection including tropism.

Table 3. The number of infection and lifetime of viruses

	$Size_p$	$\#_i$	#s	%s	$Life_v$
Virus GA	1000	9518	3236	34	0.9
	2000	39249	13143	33	0.9
Virus with tropism GA	1000	4872	1901	39	1.2
	2000	20748	8372	40	1.2

5. DISCUSSIONS

Figure 8 indicates that it is possible to implement a search technique that is robust against noise by using the cooperative action of a wide area search using GA and a local search using viruses grafted onto the chromosomes of the host entities. In this figure, there was no difference between standard GA and GA with viral grafting up to 20 generations, but a large difference in error values occurred after approximately 20 generations. This is thought to be because the wide area search function of GA works effectively at the initial stages of searching, so no particular difference is observed between the two techniques, while after the search has progressed to a certain level, the effects of noise

increase the importance of local search capabilities, so that the local search effects of parasitic viruses begin to become apparent.

From Fig. 9 it can be seen that GA with the simple addition of a parasitic virus function has more search capabilities with just 1000 entities as standard GA with 3000 entities. Specifically, in the case of problems where there is a need for local search capabilities that take noise into consideration, it can be thought that GA with viral grafting is an effective way of reducing the number of entities needed to search for a solution. When a tropism function is introduced, it can be seen that even when the number of host entities is reduced to 1000, it is possible to achieve similar search capabilities to the case of GA with viral grafting with 2000 entities. The addition of a tropism function thus allows further reductions to be made in the number of entities needed to search for a solution.

According to Fig. 10, which presents a detailed comparison of the effects of different numbers of entities, it can be seen that the overall trend of standard GA and virus infection GA with tropism is for errors to decrease as the number of entities increases and as the number of generation increases. It can also be seen that a viral infection GA with tropism has clearly more search capability than standard GA. *Z*-axis in Figure 11 means the deviation for error of the best individual at each execution. As shown in Figure 11, standard GA increases deviation, going up to down sharply. While GA infected virus with tropism increases it smoothly.

Table 3 shows the results of additional experiments performed to investigate the origins of the effects of tropism in viral infection. This table shows the results of investigating the number of infections and lifetime of viruses for each number of host entities when using viruses both with and without tropism functions. In this table, $Size_p$ is the population size, $\#_i$ is the total number of infections, $\#_s$ represents the number of successful infections (the evaluation value of a host increases due to infection by viruses), $\%_s$ represents the success rate of infections, and $Life_v$ represents the average lifetime of a virus expressed as a number of generations. As these results show, a virus that exhibits tropism tends to have a higher infection success rate, and consequently its average lifetime is about 30% longer than that of a virus with no tropism. In the both the virus entity and the host entity, tropism appears to reduce the incidence of disadvantageous infections and increases the survival rate of viruses, thereby allowing solutions to be searched for with fewer host entities.

The idea of incorporating viral evolution into GA has already been the subject of several studies [17, 19]. A fundamental difference between our proposed technique and the conventional approach is that the latter is chiefly aimed at performing a wide-area search efficiently with both GA and viruses, whereas our technique establishes a clear division of roles between GA, which is used for wide-area searching, and viral evolution, which is used for local searching. We achieve this by introducing tropism into the virus infection mechanism so as to avoid the local search of the virus impeding the wide-area search of GA. Further studies are needed with regard to the definition of viral infection for improving local search capabilities and the method used to implement tropism in order to improve the infection rate. In the future it will also be necessary to compare the proposed approach and GA with conventional local search methods.

6. CONCLUSION

We have proposed a technique for implementing genetic algorithms where viruses are grafted onto part of the chromosomes in order to improve the local search capabilities when genetic algorithms are applied to the analysis of noisy time-series signals. Specifically, we have proposed a search technique that is robust against noise by causing cooperative action between a wide area search using host chromosomes and a local search using viruses grafted onto these chromosomes. We have also improved the infection rate by providing the viruses with a tropism function. By conducting simulated evaluation experiments, we have shown that this technique is effective at improving the convergence rate of solution searches in genetic algorithms with parasitic viruses, and that by conferring a tropism function to the viruses it is possible to increase the virus survival rate, resulting in robust characteristics with respect to reductions in the number of host entities.

7. REFERENCES

[1] Tamaki, H., Kita, H., and Kobayashi, S. Multi Objective Optimization by Genetic Algorithms: A Review. In *Proceedings of the 3rd. IEEE Intl. Conf. on Evolutionary Computation*, 1996, 517-522.

[2] Parmee, I.C., and Watson, A. H. Preliminary Airframe Design Using Co-Evolutionary Multiobjective Genetic Algorithms. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, 1999, 1657-1665.

[3] Blumel, A.L., Hughes, E.J., and White, B.A. Fuzzy Autopilot Design Using A Multiobjective Evolutionary Algorithm. In *Proceedings of the 2000 IEEE Congress on Evolutionary Computation,* 2000, 54-61.

[4] Oakley, H. Two Scientific Applications of Genetic Programming: Stack Filters and Non-Linear Equation Fitting to Chaotic Data. in K.E. Kinnear, Jr (ed.), *Advances in Genetic Programming*, The MIT Press, 1994, 369-389.

[5] Sato, Y., and Nagaya, S. Evolutionary Algorithms that Generate Recurrent Neural Networks for Learning Chaos Dynamics. In *Proceedings of the 1996 IEEE Intl. Conf. on Evolutionary Computation*, 1996, 144-149.

[6] Yoshihara, I., Aoyama, T., and Yasunaga, M. GP-Based Modeling Method for Time Series Prediction with Parameter Optimization and Node Alternation. In *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, 2000,1475-1481.

[7] Reynolds, C.W. Evolution of Obstacle Avoidance Behavior: Using Noise to Promote Robust Solutions. In K.E. Kinnear, Jr (ed.), *Advances in Genetic Programming*, The MIT Press, 1994, 221-241.

[8] Beyer, H-G., and Arnold, D.V. Fitness Noise and Localization Errors of the Optimum in General Quadratic Fitness Models. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, 1999, 817-824.

[9] Beyer, H-G. *Noisy Optimization with Evolution Strategies*. Kluwer Academic Publishers, 2002.

[10] Tominaga, D., Koga, N., and Okamoto, M. Efficient Numerical Optimization Algorithm Based on Genetic Algorithm for Inverse Problem. In *Proceedings of the 2000*

Genetic and Evolutionary Computation Conference, 2000, 251-258.

[11] Goto, R., and Sato, Y. Motion Analysis of Moving Objects with Genetic Algorithms. In *Proceedings* of *the 2001 Genetic and Evolutionary Computation Conference*, 2001,1276-1283.

[12] Goto, R., and Sato, Y. Applicability of Genetic Algorithms to Motion Analysis of a Moving Object. In *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, 2002, 765-770.

[13] Goto, R. and Sato, Y. The Analysis for the Characteristics of Time Series Signals with Genetic Algorithms. In *Proceedings of the 2004 International Conference on Computing, Communications and Control Technologies*, Volume I, 2004, 267-272.

[14] Goto, R., and Sato, Y. The Motion Analysis of a Moving Object in Sea by Analyzing Doppler Effects of Sound with Genetic Algorithms. In *Proceedings of the 2006 IEEE Inter. Conf. on Systems, Man and Cybernetics*, 2006, 984-991.

[15] Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[16] Nakahara, H., and T. Sagawa, T. *Virus Evolution Theory*. Tairyu press, 1989.

[17] Kubota, N., Hukuda, T. Scheme Representation in Virus Evolutionary Genetic Algorithm for Knapsack Problem. In *Proceedings of the 1998 IEEE Intl. Conf. on Evolutionary Computation*, 1998.

[18] Tyler, K. L. and Fields, B. N. Pathogenesis of viral infections. In *Fields Virology*, third edition. (Fields, B. N., Knipe, D. M., Howely, P. M. et al. eds.), Lippincott-Raven Publishers, Philadelphia, 1996, 173-218.

[19] Kubota, N., Hukuda, T., Shimojima, K. Trajectory Planning of Cellular Manipulator System Using Virus-Evolutionary Genetic Algorithm. *Robotics and Autonomous Systems 19*, 1996, 85-94.