Real-time adaptive maintenance for performance improvement on daily-use computer

Shinichi Hirose
Graduate School of Information Science and Technology,
Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido,
060-0814, Japan
E-mail: s.hirose@complex.eng.hokudai.ac.jp

Shinichiro Yoshii
Graduate School of Information Science and Technology,
Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido,
060-0814, Japan
E-mail: yoshii@complex.eng.hokudai.ac.jp

Abstract

When it continues using a personal computer, it has often come to operate slowly rather than the time of having purchased it. In this paper, the authors argue about the easy tuning method of personal computers. The general users without much deep special knowledge will be able to use it easily. By using the simple genetic algorithm, we search for the optimized set of the parameters of the operating system. the effect is verified in an experiment. The authors propose the adaptive method for the improvement in performance for personal computers.

1. Introduction

It has been long time since the personal computers turned to be daily commodities. As being used for a long time, the personal computer comes to lose its quickness. For example, it comes to take a long time to boot up. So the personal computer users have often come to feel inconvenience for use. The most of general users do not have deep knowledge as much as IT professionals. Therefore, general users cannot help making a uniform improvement in performance by installing the packaged software for tuning, or returning their personal computers to the initial state by installing the operating system again. However, since various factors for the prolongation of the booting up time and the other inconvenient phenomena can be considered, these methods do not necessarily become better solution.

Several system integration companies have proposed the self-control function in the information systems called Autonomic Computing in enterprise field [1]. But it is expected that the maintenance method of personal computers as daily commodities is more adaptive than the packaged software for tuning and simpler than Autonomic Computing. The authors focused on tuning parameters of the operating system as the maintenance method of personal computers.

2. Genetic algorithm

Genetic algorithms bases on the research of J. H. Holland [2] in University of Michigan in the 1960s. He made the model in engineering paying attention to the genetical mechanism in a nature. The algorithms based on such a view have been studied as various and concrete techniques. The most fundamental kind of genetic algorithm in them is called simple genetic algorithm [3]. Hereinafter, we explain the simple genetic algorithm .

A certain individual is formed based on the genetic code described in its chromosome. The chromosome is constituted as a gene arrangement with which two or more genes were located in a line, and it is decoded when an individual is born. The character of the individual decoded and generated is called phenotype. On the other hand, the pattern of the gene of the origin of it is called genotype. In genetic algorithms, it is usually assumed that the individual has only one chromosome. The gene information will be recorded by the following arrangement S in the individual with n genes.

$$S = \langle s_1, s_2, \cdots, s_n \rangle, \qquad s_i \in \{0,1\}$$

The position of each gene on a chromosome is called locus, and the gene in the position specifies a certain characteristic character. For example, s_2 is recording the information on the color of the eye of an individual etc. The candidate of the gene that can be taken on a locus is called allele. As mentioned above, an individual will be born to environment by decoding and expressing a genetic code from a chromosome. And a set of an individual is called population. The individual with the high degree of adaptation to the environment survives by high probability. The evaluation value is called fitness.

Generally the genetic algorithm is advanced by the following procedures. When solving an optimization problem, it is necessary to express a problem to optimize with a gene and to set up end conditions.

(1) initialization : Building the individuals by generating their chromosomes randomly
(2) evaluation : Calculating the fitness for each individual according to the evaluation function
(3) selection : Selecting the individuals by their fitness
(4) crossover : Generating the chromosomes of the new individuals by crossover two parents' chromosomes
(5) mutation : Generating the chromosomes of the new individuals by the probability of mutation
(6) reproduction : Replacing with all or some of new generations
(7) repetition of (2) to (6) until fulfilling end conditions

3. Experiment

Generally, that users feel some inconvenience to computer systems becomes the opportunity that tunes

them up. In many cases, after the tuning target is set up, a method required in order to attain the target is examined. The tuning is performed according to the plan. The phenomenon which user feels inconvenient is various. There may be two or more tuning methods for the same phenomenon.

3.1 Tuning method

In order to improve the prolonged startup time of the personal computer, we search for the optimized set of parameters of the operating system by the simple genetic algorithm. In this experiment, Microsoft Windows XP, which is the most popular operating system among general users, was taken up. The parameters of operating system of Microsoft Windows, which are called registry, can be set with the numerical values other than 0 or 1, or character strings, or binary data, etc. The authors selected the 14 target parameters, which can be set with 0 or 1, taking account of their roles with expertise acquired through professional practice of system engineering. We coded to genotypes the set of parameters chosen in this way. Instead of using the defined evaluation function, we used the starting time actually obtained by measurement in the set of parameters as each individual's evaluation. In order to measure the starting time of a personal computer, we referred the document [4] that provided guidelines for system manufacturers to improve boot and the tool introduced there.

We summarized some main genetic operators and parameters of the simple genetic algorithm for our experiment in Table 1.

	GA operators	GA parameters
(1)	initialization	N (population of individuals) = 30
(2)	selection (elitist preserving selection)	G (generation gap) = 0.8
(3)	crossover (uniform crossover)	p_c (probability of crossover) = 0.8
(4)	mutation	m_c (probability of mutation) = 0.01

Table 1. The genetic operators for the experiment

3.2 Experiment System

Figure 1 shows the experiment environment. We experimented on the virtual operating system that was realized by this emulation software.

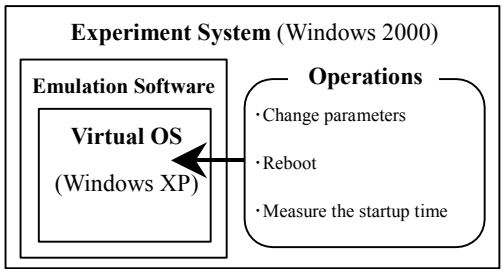

Figure 1. The experiment environment

This emulation software had the function that rollbacks the operation to the virtual operating system, and we used this one when acquiring an individual's evaluation. Because the internal environment of the operating system is changed by the repeating configuration of the registries, it is impossible to acquire the exact starting time of the personal computer.

3.3 Results

We repeat the reboot of the virtual operating system supposing the situation that the personal computer was used daily. In the case A and B, after rebooting the virtual operating systems 100 times and 300 times respectively, we measured the start time without tuning parameters. Similarly, in the case A' and B', after rebooting the virtual operating systems 100 times and 300 times respectively, we measured the start time with tuning parameters. Table 2 shows the results. When a case A and A' were compared, 9.58% of improvement was found, and when a test case B and B' were compared, 16.52% of improvement was found.

	Boot Times	Startup time [s]	effect
Case A	100	50.64	—
Case A'	100	45.79	△ 9.58 %
Case B	300	68.81	—
Case B'	300	57.44	△ 16.52 %

Table 2. The experiment of the results

4. Conclusion

We consider that a certain amount of improvement was found.

In an experiment of this paper, it is hardly taking into consideration that the state of the hardware and software comes to change with using the personal computer. The remaining issues will be to build the model that realizes the method of tuning up the value of the parameters with taking into account the change state of hardware environment, and to check it in the experiment.

References

[1] IBM Corporation, Autonomic Computing, http://www-3.ibm.com/ autonomic/index.shtml, 2004.
[2] J. H. Holland, Adaptation in Natural and Artificial Systems, MIT Press 1992.
[3] M. Sakawa and M. Tanaka, Genetic Algorithm, Asakura-Syoten, 1995.
[4] Microsoft Corporation, Fast System Startup for PCs Running Windows XP, http://www.microsoft.com/whdc/ system/sysperf/fastboot/fastboot-winxp.mspx, 2002.