2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

An Extended Hybrid Genetic Algorithm for Exploring a Large Search Space

Hong Zhang and Masumi Ishikawa
Graduate School of L.S.S.E., Kyushu Institute of Technology,
2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan
{zhang, ishikawa}@brain.kyutech.ac.jp

Abstract

Recently, a hybrid methodology for combining genetic algorithms and local search algorithms has received considerable attention. This paper proposes an extended hybrid genetic algorithm to further improve the performance of finding the optimal solution in a large search space. Three key ideas, i.e. the *elitism*, non-redundant search, and steepest-ascent hill climbing, are introduced into a standard genetic algorithm. The first one is to copy superior individuals to the next generation for improving convergence. The second one is to increase the efficiency of finding the best individual, and the third one is to increase the efficacy of finding the best individual. Through the combination of these ideas, the proposed method is well suited to find the optimal solution in a large search space corresponding to a given problem. To evaluate the effectiveness of the proposed method, computer experiments that estimate the weights of connections in neural networks for solving XOR problem are carried out. The results well demonstrate its effectiveness.

Keywords: hybrid genetic algorithm, non-redundant search, steepest-ascent hill climbing, learning, neural network .

1 Introduction

Genetic algorithms (GAs) proposed by J.H. Holland [3], which use evolutionary computation, have been applied to real world problems in many application disciplines. Recent studies have shown its effectiveness over conventional search and optimization methods for solving complex and ill-posed problems. In spite of the simplicity of operators used, the standard genetic algorithm (SGA) [1] is good at solving complex optimization problems [4].

Although the SGA has advantages of adaptability and robustness, it still has disadvantages such as no guarantee for convergence and instability in computation. These greatly affect the ability of finding the optimal solution in a large search space. To improve the performance of the SGA, heuristic algorithms have been introduced into the SGA [5, 6, 7]. Although these approaches are effective, they are not suited to combinatorial optimization problems with time-varying parameters due to heavy computation.

In previous work, we proposed a hybrid genetic algorithm (HGA) as a kind of global search [8]. The point includes adoption of the *elitism* for improving convergence and a non-redundant search for quickly finding the optimal solution in a large space. Due to the simplicity of operators used such as selection, comparison, and conservation, the HGA is well suited to quick acquisition of a sequence of optimal solutions corresponding to the changing environment.

Many computational and cognitive tasks involve a search in a large space. It is well known that GAs are good at global search, but weak at local search. For improving the performance of finding the optimal solution, we propose to extend the HGA by incorporating local search.

Hill-climbing restricts the search around the best solution found so far by exploring its direct neighbourhood [10, 11]. Although hill-climbing has a drawback which has inability to detect the unsolvability of a problem instance, it provides a powerful strategy for exploring the optimal solution. In addition to this, local search requires less working memory, and is easy to implement. As an example, hill-climbing technique has been applied to SGA for solving a toy optimization problem [12].

Based on these characteristics of hill-climbing, we expect the extended hybrid genetic algorithm to reinforce the efficacy of finding the optimal solution in a large search space by performing global search and local search simultaneously.

The rest of the paper is organized in the following way: Section 2 describes steepest-ascent hill climbing, ranking, and non-redundant search algorithms. Section 3 gives a flow chart of the proposed algorithm and its features. Section 4 provides results of computer experiments, which demonstrates the effectiveness of the proposed method. Section 5 concludes the paper.

2 Algorithms for improvement

2.1 SAHC algorithm

Exhaustive local search algorithm provides a local optimal solution in the neighbourhood of an individual. We, here, refer to the algorithm as steepest-ascent hill climbing (SAHC). The procedure of the SAHC is as follows.

Step 1: Select an initial individual, **x**, randomly. We refer to the individual as a core one.

Step 2: Generate the neighbourhood individuals of **x**, **y**, with Hamming distance, $d_H(\mathbf{x}, \mathbf{y})$, less than a threshold. For clarity, Figure 1 shows the set of neighbourhood individuals of **x**={0000} with d_H=1 and 2.

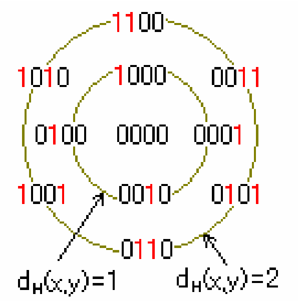

Figure 1: The core individual, **x**={0000}, and its neighbourhood individuals.

Step 3: Calculate the fitness of each neighbourhood individual. Let the individual with maximum fitness value be the core individual, **x**.

Step 4: Repeat **Step 2** through **Step 3** until the core individual is not changed.

The SAHC algorithm can find out the best individual in the neighbourhood by a series of comparison within the set of neighbourhood individuals.

2.2 Ranking algorithm

Instead of the roulette wheel selection, we use the *elitism* for improving convergence. Accordingly, the individuals in the current population are sorted according to their fitness. The sorted population is referred to as a new one. Then the index of individual in the population will be used to determine which individuals are allowed to reproduce directly to the next generation.

2.3 NRS algorithm

Non-redundant search (NRS) is a kind of global search. The basic construction of its algorithm is as follows. First action is deletion of individuals with the same chromosome in the current population. Second action is addition of new individuals selected randomly instead of these redundant ones. Consequently, non-redundant search improves the efficiency of search for finding the optimal solution by the expansion of search range.

3 Proposed method

By carrying out global search (NRS) and local search (SAHC) simultaneously, we can avoid the problem that falls into local minimum, and keep the diversity of individuals in search process to realize an efficient search. Figure 2 shows the flow chart of the extended hybrid genetic algorithm (EHGA) which improves convergence and increases the capability of finding the optimal solution.

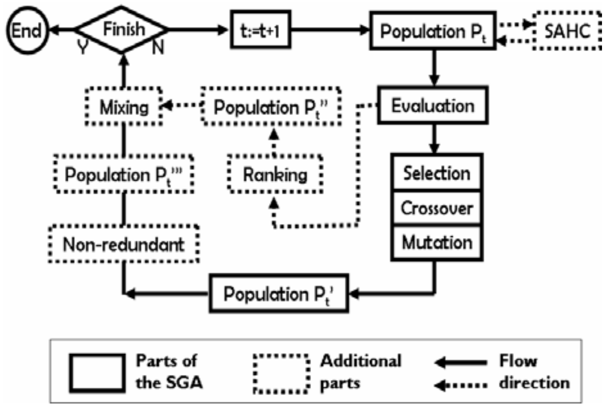

Figure 2: Flow chart of the extended hybrid genetic algorithm.

The EHGA uses the following operators in addition to those in the SGA.

Operator 1: Select some individuals in population P_t, and determine the best individual in their neighborhood set corresponding to core individual through the SAHC algorithm.

Operator 2: Adjust order of the individuals in the current P_t based on their fitness. Let the ordered population be P_t''.

Operator 3: Delete redundant individuals from the population P_t', which are generated by the SGA and add new individuals selected randomly. Let the resulting population be P_t'''.

Operator 4: Merge the number of the best individuals from the population P_t'' and the number of obtained individuals from the population P_t''' in a certain proportion for generating the next generation P_{t+1}.

4 Computer experiments

To demonstrate the effectiveness of the proposed method, we discuss how to estimate the weights of connections in neural networks for solving the exclusive-OR (XOR) problem, well-known problem [9] by the EHGA.

4.1 Expression of neural network

Figure 3 indicates a structure of neural network with two inputs, two hidden-layer units, and a single output unit for solving XOR problem in which four inputs of {0, 0}, {0, 1}, {1, 0}, and {1, 1} must be mapped to outputs of 0, 1, 1, and 0, respectively.

Each of the three units (2-hidden, 1-output) has three weights (one of them represents the bias term of the unit). And each unit is nonlinear one which has the standard logistic output function, $f(z) = 1/(1+E^{-z})$, where z is net input of the unit, that is the sum of an adjustable bias term, w_0, and dot product of the incoming activations and their associated weights, i.e. $z = \sum_{i=0}^{2} w_i x_i, \quad x_0 = 1.$

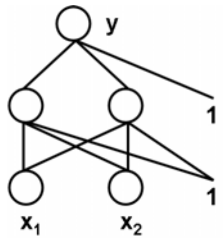

Figure 3: A structure of neural network for solving XOR problem.

We, here, represent each weight of connections in the neural network by a m-bit binary string, then scaling the weight to a number between -5 and 5. Finally, concatenate all of the weights together into one long string which length is 9x m-bit, $\mathbf{w}=\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}=\{w_{11}, w_{12}, w_{10}, w_{21}, w_{22}, w_{20}, w_{31}, w_{32}, w_{30}\}$ as one individual for evolutionary computation. Notice that $\mathbf{w}_1$ and $\mathbf{w}_2$ denote the weights of hidden units, and $\mathbf{w}_3$ denotes the weights of output unit.

We use directly the value that is the inverse of the sum of mean squared error (MSE) of the neural network outputs and a constant as the objective function. Hence, the following fitness function, $g_k(\mathbf{w})$, is considered to evaluate the individual of the weight of connections in *k*th neural network,

$$g_k(w) = \frac{1}{MSE_k + \mu} = \frac{1}{\frac{1}{4}\sum_{i=1}^{4}(t_i - y_i)^2 + \mu} \qquad (1)$$

where t_i and y_i is the target output and the actual output according to the weights of connections in *k*th neural network, respectively. μ is a minute constant for preventing the denominator of Equation (1) becomes zero.

4.2 Experimental results

In our computer experiments, the major parameters of the genetic algorithms applied to solve XOR problem are as follows.

Population size: 100; the number of generations: 100; selection: roulette wheel selection; crossover: 6-point crossover (p_c=0.8); bit-wise mutation (p_m=0.1).

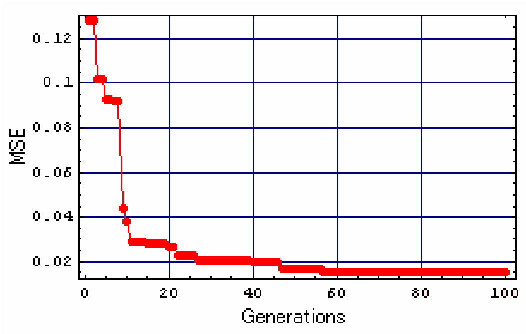

Figure 4: The MSE of the neural network outputs corresponding to each generation for solving XOR problem.

For computational convenience, Make m-bit=6 binary string to represent each weight of the neural network in Figure 3. Nevertheless, we need to find the optimal solution (best weights) in the large search space, the order of 2^{54} points, for solving XOR problem.

Computer experiments are implemented. As a result, Figure 4 shows a convergence process of the neural network corresponding to each generation for solving XOR problem by the EHGA.

From Figure 4, it is clear that the approximate perfect mapping function is evolved within 100 generations allowed. The estimated weights of each unit of the neural network are: $\mathbf{w}_1$={5., -5., -2.9365}; $\mathbf{w}_2$={-5, 5, -2.9365}; $\mathbf{w}_3$={5., 5., -2.4603}. Figure 5 shows the curved plane determined by these weights for distinguishing 0/1 classes of XOR problem.

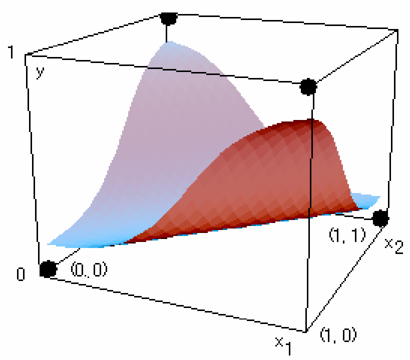

Figure 5: The curved plane for solving XOR problem.

To compare the performance of the proposed method and other algorithms and confirm a role of the parameters such as Nc, Ns (Nc is the number of copied superior individuals to the next generation, and Ns is the number of selected individuals applied to the SAHC algorithm.) and so on, Figure 6 indicates the histograms of the results which reflect probabilities related to fitness distribution obtained.

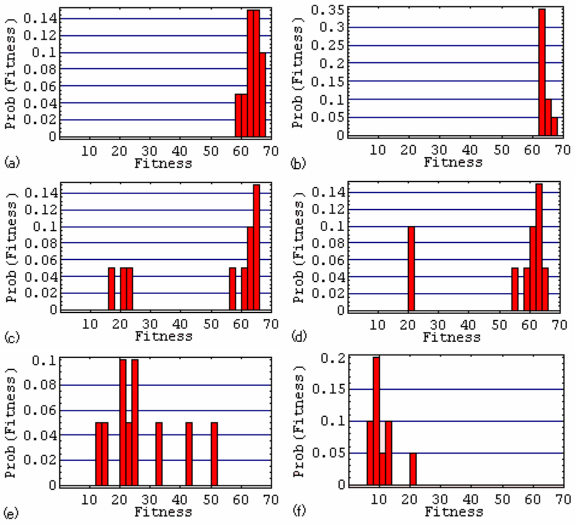

Figure 6: Probabilities related to fitness distribution. (a) The EHGA with Nc=1, Ns=3, and d_H=2. (b) The EHGA with Nc=1, Ns=1, and d_H=2. (c) The EHGA with Nc=10, Ns=10, and d_H=1. (d) The EHGA with Nc=1, Ns=10, and d_H=1. (e) The SGA with Nc=10. (f) The SGA.

Obviously, in comparison with the result of the SGA in Figure 6(f), the result of the SGA with the *elitism* in Figure 6(e) is better for solving XOR problem. However, it cannot find much better weights as stated above than the results of the EHGA in Figure 6(a) and Figure 6(b).

Furthermore, through comparing the results between Figure 6(a) and Figure 6(b), we can confirm the fact, i.e. the greater the number of selected individuals is, the bigger the probability of the obtained best fitness is.

For examining the role of parameters in the EHGA, Table 1 indicates the statistic results corresponding to fitness in the different cases.

Table 1: The maximum, mean, minimum values of fitness in the different cases.

Case	Maximum	Mean	Minimum	Conditions
(a)	66.4291	63.7045	58.2817	Nc=1, Ns=3, d_H=2
(b)	66.4291	63.9784	62.4272	Nc=1, Ns=1, d_H=2
(c)	65.5429	50.0523	16.7424	Nc=10, Ns=10, d_H=1
(d)	65.2093	53.4165	20.9490	Nc=1, Ns=10, d_H=1

From the results of Table 1, it is clear that the larger the neighbourhood range taken is, the bigger the probability of the obtained best fitness is.

These results indicate that the SAHC algorithm is very powerful for exploring the large search space, and provide evidence of the efficacy for finding the optimal solution by the proposed method.

5 Discussion

In fact, as we know that solving XOR problem itself is not a big search problem. However, due to satisfying higher estimate accuracy to solve XOR problem, a binary-coded GAs has to explore a large search space which represents the weights of connections in the neural network for searching the optimal solution.

For reducing the search space in scale, a simple solution to the problem is the use of floating-point representation of the parameters as an individual. In real-coded GAs [2, 13], an individual is coded as a finite-length string of the real number corresponding to each weight of connections in the neural network. We, here, make a concession about the expansion to real-coded GAs of the proposed method.

6 Conclusions

We have proposed an extended hybrid genetic algorithm with three strategies, i.e. the elitism, non-redundant search and steepest-ascent hill climbing. Based on the combination of advantages of three key points applied to it, the proposed method further improves the efficiency and efficacy for finding the optimal solution in a large search space.

Applications of the proposed method to estimate the weights of the neural network for solving XOR problem well demonstrate its effectiveness. Obtained results indicate that the EHGA performs better than the other approaches such as the SGA and the SGA with the *elitism*.

Because of the given problem, XOR problem, in our computer experiments is quite simple, and due to enforcement of the SAHC algorithm, shortening its computational time to be necessary, it is left for further study to apply the proposed method to real world problem.

7 References

[1] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Boston, USA (1989).

[2] Goldberg, D.E., ''Real-coded genetic algorithms, virtual alphabets, and blocking,'' Complex Systems, No.5, pp.139-167 (1991).

[3] Holland, J.H., *Adaptation in natural and artificial systems*, Reprinted, MIT press, Cambridge, USA (1992).

[4] Man, K.F., Tang, K.S. and Kwong S., *Genetic Algorithm*, Springer-Verlage, London, UK (1999).

[5] Thierens, D., "Scalability Problems of Simple Genetic Algorithms," *Evolutionary Computation* 7(4): pp331-352 (1999).

[6] Ito, M. and Sugisaka, M., "A Study of Selection Operator Using Correlation between Individuals in Genetic Algorithm," *IEEJ Trans. EIS* 124(1): pp.170-175 (2004) (in Japanese)

[7] Whitley, D. et al., "Scheduling Problems and Travelling Salesman: The Genetic Edge Recombination Operator," Proceedings of the 3rd International Conference on Genetic Algorithms, pp.133-140 (1989).

[8] Zhang, H. and Ishikawa, M., "A solution to combinatorial optimization with time-varying parameters by a hybrid genetic algorithm," *Proceedings of international Symposium on Bio-inspired Systems (BrainIT2004)*, in press.

[9] Rumelhart, D.E. et al., Parallel Distributed Processing, The MIT press, Cambridge, USA (1986).

[10] Russel, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, NJ, USA (1995).

[11] Lin, S. and Kernighan, B.W., "An effective heuristic for the travelling-salesman problem," *Operational Research* 21: pp498-516 (1973).

[12] Prugel-Bennett, A., "When A genetic Algorithm Outperforms Hill-Climbing," *Theoretical Computer Science* 320(1): pp.135-153 (2004).

[13] Tsutsui, S. and Goldberg, D.E., ''Search Space Boundary Extension Method in Real-Coded Genetic Algorithms,'' *Information Sciences*, Vol.133/3-4, pp.229-247 (2001).