

Island Model GP with Immigrants Aging and Depth-Dependent Crossover

Makoto Iwashita Hitoshi Iba
Graduate School of Frontier Science
University of Tokyo
7-3-1 Hongo Tokyo, Japan 113-8656

Abstract -

This paper proposes a new method for island model GP. The proposed method applies a traditional genetic operator to an aborigine and a depth-dependent crossover to the immigrants according to their ages, which show how long they survive in the island. This method can provide both local and global search strategies. The experimental results have shown that our approach works effectively.

I. Introduction

According to the building block hypothesis and Koza's schema theorem, one of the reasons why GP works effectively is that small building blocks are combined by crossover to construct a bigger building block [1]. Thus, for the purpose of effective search in GP, both local search strategy, which can create new schemata and can optimize schemata, and global search strategy, which can combine schemata into bigger building blocks, have to be provided.

Island model is a coarse-grained parallel strategy for evolutionary algorithms. In the island model, a population is divided into subpopulations (demes), and some of the individuals are exchanged periodically (i.e. migration), as shown in Fig.1. In this method, each deme can grow its original schemata and global search ability is expected to be provided by combining the schemata of different demes by crossover. Tanese demonstrated that the island model GA has the ability to find fitter individuals than the traditional GA [2]. Also, other papers reported the superiority of the island model GA and GP [3]-[5]. For instance, Miki et al adopted a new method that provides high local search ability in each subpopulation in the island model GA, and succeeded in establishing more effective search [6]. Similar result is expected by providing both local and global search ability in the genre of island model GP.

In the tree-expressed GP, it is considered that there is a relationship between the locality of the search and the depth of the node to which a genetic operation is applied. Rosca and Ballard hypothesized that the farther away the genotypic change occurs from the root node (i.e. the depth of crossover point is large), the smaller is the corresponding change in the phenotype [7]. Igel et al

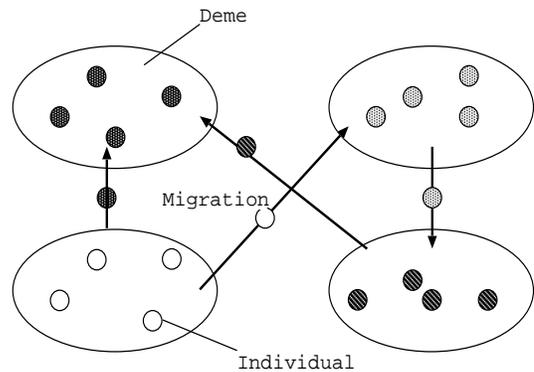


Fig. 1. Island Model

showed that this hypothesis is valid for selected test problems experimentally [8]. In the normal crossover of GP, the node selection ratio, i.e. the probability of selecting a node as a crossover point, is chosen to be uniform. Since the number of nodes of depth d tends to increase with d , the crossover point becomes deep, especially when the tree has grown to be a large size. Thus, GP with normal crossover tends to incline on the local search as the tree grows.

Ito et al proposed a new method, i.e. the depth-dependent crossover, to overcome this problem [9]. With this method, the depth selection ratio for a crossover is higher for a node closer to the root node (i.e. crossover point tends to be shallow). According to the Rosca and Ballard's hypothesis, this method puts emphasis on global search. They stated that this method works as protection against destructive crossover and accumulates building blocks, but it generated very large programs. In their experiment, the effectiveness of this method was proved to be problem-dependent.

According to Rosca and Ballard's hypothesis, the normal crossover is inclined on local search ability, whereas the depth-dependent crossover is inclined on global search ability. Hence, we consider that the combination of these two would work effective, especially in the island model GP, in which aborigines and immigrants have different roles. We propose a new index, i.e. immigrant age, which indicates how long an individual has survived in the deme.

Our approach is to apply a crossover method to the individual's immigrant age. The effectiveness of the proposed method is empirically shown by several experiments.

II. Method

A. Immigrant age

In the island model GP, different schemata grow on each deme. Thus, schemata of newly migrated individuals and their descendants are considered to be rare in the deme. Therefore, it would be useful to discriminate those new individuals from the others. We propose a new index, i.e. *immigrant age*, that shows how long an individual has survived in the deme. The definition of immigrant age is described below:

- Immigrant age of an individual in the initial generation is zero.
- Immigrant age of a migrated individual is reset to zero.
- Immigrant age of an individual is increased by one when the genetic operation except crossover is applied.
- When crossover is applied, the younger age of its parents is taken as offspring's immigrant age, and is increased by one.

When running a GP application, the loss of population diversity would often be a problem that causes the decline of the searching ability. In the island model GP, immigrants and their descendants would play an important role in keeping the population diversity. We can discriminate those individuals by means of the immigrant age. We consider that the younger the immigrant age of an individual is, the more plausibly it has rare schemata which may be important for effective search.

B. Applied Operation

In the traditional method, the genetic operator to be applied is determined stochastically according to a certain ratio after the selection. However, it would not take the full advantage of the island model (even if plural methods of crossovers are used). We propose a new method for determining the operation to be applied, which is described below:

- Select an individual by using the traditional method.
- If the immigrant age of the selected individual is less than the threshold (i.e. a fixed constant), **always** apply the depth-dependent crossover and **not** apply the mutation.
- Otherwise, determine the operator stochastically as the traditional method; apply a normal crossover if needed.

By applying the depth-dependent crossover to the "deme-young" individuals, rare schemata are protected from the

destructive crossover or mutation, and rare schemata from other demes and deme-original schemata are combined to make a bigger building block is facilitated. By applying the traditional method to "deme-old" individuals, the optimization of schemata and the construction of new schemata are maintained. We consider that this method integrates both local and global search strategies, and takes full advantage of the island model.

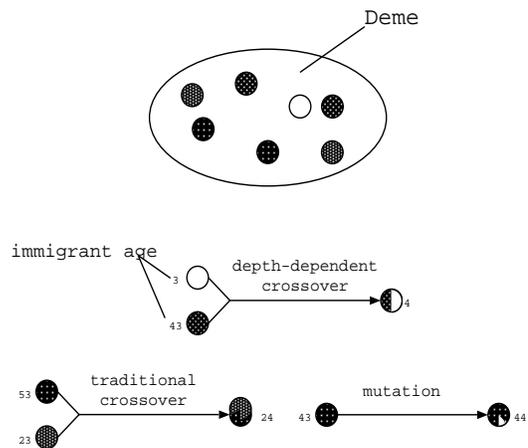


Fig. 2. Proposed method

C. Depth-Dependent Crossover

We applied Ito's depth-dependent crossover [9]. The algorithm of Ito's depth-dependent crossover is summarized below:

STEP1. Given a tree, determine the depth d for applying the depth-dependent crossover.

STEP2. Select randomly a node of which depth is equal to d in STEP1.

STEP3. Apply the canonical crossover for the nodes chosen in STEP2.

The depth selection ratio is derived by using the following equations:

$$\begin{cases} threshold_i = 1/2_i & \text{for } i = depth, \\ threshold_i = threshold_{i+1} * 2 & \text{for } i = 0, 1, \dots, depth - 1. \end{cases}$$

where $threshold_i$ is the depth selection ratio at the i th depth, and $depth$ is the depth of a tree. $threshold_i$ is an accumulated value from the deepest node. The above equations represent that the depth selection ratio is set to be 1.0 for a root node. The depth selection ratio is half of its parent node's ratio. When determining the depth d for the depth-dependent crossover, we pick up a number i between 0 and the maximum depth in proportion to the $threshold_i$ value.

In Fig.3, the depth selection ratio of a root node is 50.0% following the above definition. Note that the

threshold value of a root node in the figure is accumulated from the deepest node.

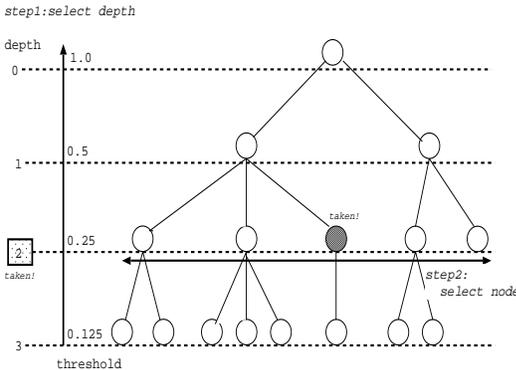


Fig. 3. Depth-Dependent Crossover

III. Experiments

A. Test problems

Three types of test problems were used to examine the performance of the proposed method; i.e. 11-multiplexer problem, ant problem, and inter-twined spiral problem.

The task of the 11-MX (11-multiplexer) problem is to decode an address encoded in binary and to return the binary data value of the register at that address [1]. An 11-multiplexer problem has 3 binary-valued address lines ($a_0 - a_3$) and 8 data registers ($d_0 - d_7$) of binary values. The fitness of the 11-MX is an error rate for total inputs. Two different function sets were used in this problem. One is the function set that Koza used, which consists of {**and**, **or**, **not**, **if**} (Koza's function set). The other is the set of logical elements, which consists of {**and**, **or**, **not**, **nand**, **xor**} (logical function set). The set of terminals used in this problem is $\{a_0 - a_3, d_0 - d_7\}$.

The ant problem is the task of navigating an artificial ant so as to find all 89 foods lying along an irregular trail on 32×32 world (Santa Fe trail) [1]. The ant's goal is to traverse the entire trail (thereby eating all of the foods) within limited energy. The fitness of the ant is the ratio of which the ant could not eat 89 foods. The function set used in this problem was {**if-food-ahead**, **prog2**, **prog3**}, and the terminal set was {**left**, **right**, **move**}.

We also experimented an extended version of this problem, i.e. the ant problem with random obstacles to test the robustness of the evolved program that GP. In this problem, an obstacle is placed randomly on the correct pathway of the problem. Foods are also added to create the guideway so as not to lose the trait of Santa Fe trail. The food-adding policy is as follows; *Place the food on the trail so that every corner of the right pathway has a food that leads the ant to the right direction.* For example, if an obstacle is placed in a straight trail, place two foods that form a parallel line with the right path (food1 and food2

in Fig.4). Add another food to lead the ant back to the original trail (food3 in Fig.4) if necessary. If an obstacle is placed on the corner of the right pathway, place one food that leads the ant to turn to the right direction (food4 in Fig.4). The difficulty of this problem depends on the place where the obstacle was placed. Therefore, the seed values for generating the random value to determine the place of the obstacle are kept at the same value over the methods. In other words, seed value of n th run is same for all the methods in comparison (of course, random value for genetic operations are not affected by this).

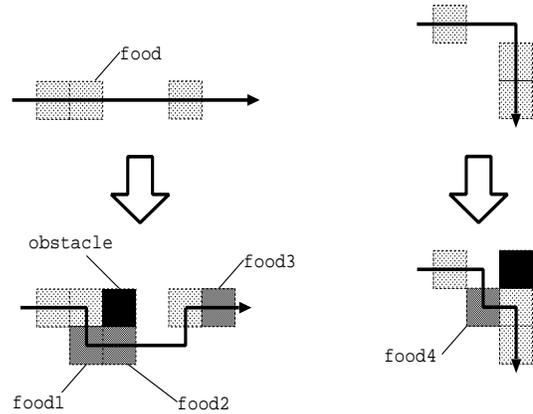


Fig. 4. Obstacles and Pathway

The task of inter-twined spirals is to distinguish two intertwined spirals [10]. The x-y-coordinates of 97 points from each spiral are given as shown in Fig.5. The problem involves learning to classify each point as to which spiral it belongs. The fitness of this problem is an error rate for the all 194 points. The function set used in this problem was $\{+, -, *, /, \sin, \cos, \tan, \log, \ln\}$, and the terminal set used was $\{x, y, \mathbb{R}\}$.

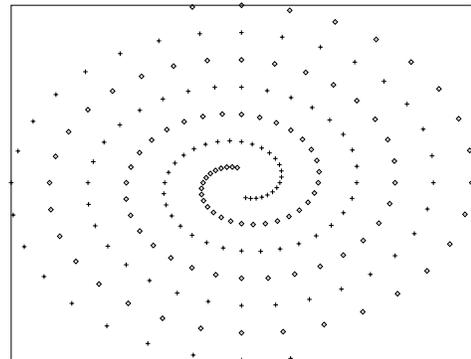


Fig. 5. Inter-Twined Spirals

B. Settings

The experiments have been executed in our PC-cluster that consists of 8 calculating nodes. Each calculating node contains AMD Athlon 900MHz as a processor and 512MB of memory. Parameter settings used in the experiments are shown in the table.

TABLE I
GP PARAMETERS

	Ant	11-MX	Spiral
No. of demes	8	8	8
Subpopulation size	512	512	512
Grow method	grow	grow	grow
Selection method	tournament	tournament	tournament
Tournament size	4	4	4
Crossover ratio	0.7	0.7	0.7
Mutation ratio	0.2	0.2	0.2
Migration freq.	10	10	10
Threshold	5	5	5

Each of the experiments has been run for twenty times using the traditional and the proposed method.

IV. Results and Discussion

The fitness graphs of all the experiments are shown in Fig.6-9. As you can see from the graphs, the proposed method outperformed the traditional method in all of the experiments, except in the 11-MX problem with Koza's function set. In 11-MX problem with Koza's function set, significant difference can not be found. For the ant problem (Fig.6) and 11-MX with logical function set (Fig.7), we can observe that the difference of performance is not that much in the forepart, but it becomes bigger as the evolution proceeds. In the initial stage, the size of trees is not so large, and the schema size is relatively small as well, so that the normal crossover works effectively. But when the trees and schemata grow large, most of the normal crossover would become destructive or neutral. This results in the stagnation of the evolution as seen in the graphs (especially clearly shown in Fig.6). On the other hand, the proposed method succeeds in combining bigger schemata to keep the evolution ongoing.

Significant difference could not be found in 11-MX with Koza's function set, while the superiority of the proposed method was evident in 11-MX with logical function set. Since Koza's function set reduces the search space and the size of the solution tree, the solution seems to have been acquired before the global search ability of the traditional method got worse.

In the fitness graph of the ant problem with random obstacles (Fig.8), we can see that two methods performed similarly until about 170th generation. However, while the fitness keeps on fluctuating with almost the same amplitude for the traditional method, the proposed method resulted in lowering the fitness values with generations. In other words, the proposed method succeeded in acquiring

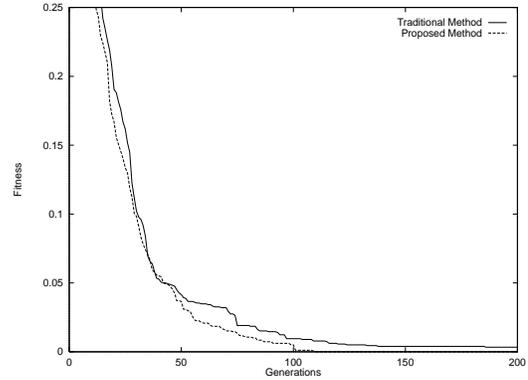


Fig. 6. Fitness vs. Generation (Ant Problem)

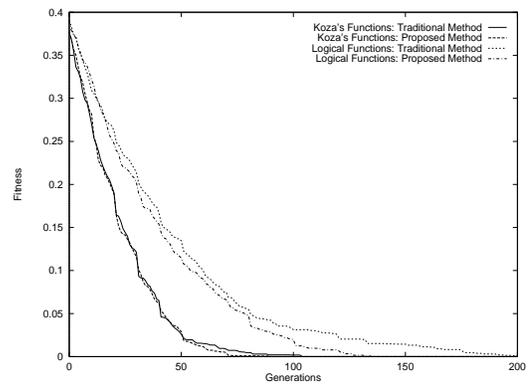


Fig. 7. Fitness vs. Generation (11-MX)

more robust individuals.

Generations when a solution was acquired for the ant problem and the 11-MX problem with logical functions are shown in the TABLE II. It is interesting that the superiority of the proposed method is clear for the average and worst generations, while no significant difference can be found for the best generations. The influence of bloat might give a good explanation to this fact. When the ability of search is reduced, trees tend to bloat, so as to grow the introns since most operations work destructively. The bloat itself also causes the reduction of the search ability, increasing the possibility of neutral operations and falling in vicious cycle. In some runs, solutions were acquired "luckily" before falling in this cycle, but in other cases the search ability was reduced spirally.

Superiority of the proposed method is more clearly shown when the success generation are compared. For example, for the ant problem, the proposed method was successful in acquiring the right solution within 100 generations in 90% of the runs, and even the worst run only needed 110 generations. On the other hand, the traditional method succeeded only 60% within 100 generations, and failed with 10% even after 300 generations.

By these experiments we can confirm that the proposed method performed effectively, especially in the difficult

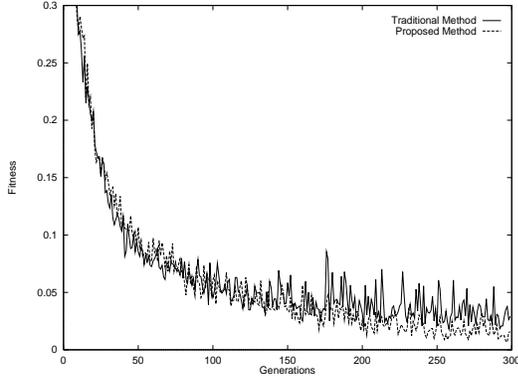


Fig. 8. Fitness vs. Generation (Ant Problem with random Obstacles)

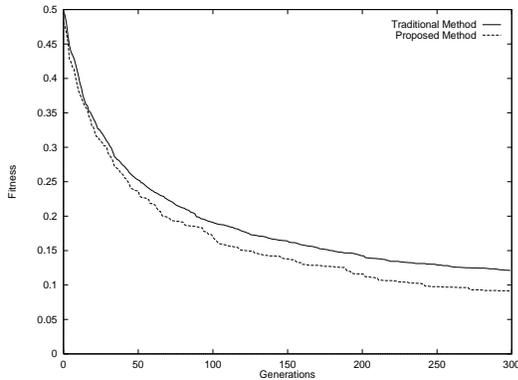


Fig. 9. Fitness vs. Generation (Inter-twined Spirals Problem)

problems.

V. Verification

The proposed method performed better than the traditional method in our experiments. In this section, we try to verify the basis of it experimentally. For this purpose, we have done another experiment to examine how "decent individuals" are created in the island GP. The definition of a "decent individual" is the individual fitter than any of the individuals of the previous generations in the deme. Immigrants are excluded from the definition. Thus, a decent individual is created only when the crossover or the mutation operators work effectively.

We selected 11-MX with logical function set as a target problem. In this experiment, except for reproducing 1% of individuals as elites, either crossover or mutation is applied at the same probability to produce the individuals for the next generation (i.e. both of the probabilities of crossover and mutation are set to be 50% each). This experiment has been run for 10 times, and we took the following statistics for all the decent individuals:

- Applied genetic operation.
- The depth-ratio to which the operation has been ap-

TABLE II
SUCCESS GENERATION

	Best	Worst	Average
11-MX L ¹ Trad. Method	74	241	130.9
11-MX L ¹ Prop. Method	80	135	103.1
Ant Trad. Method	32	N.A.(in 300)	90.9 ²
Ant Prop. Method	22	110	55.2

¹L stands for logical function set

²Average of the runs that acquired the solution within 300 generations

plied (this value would be 0 if the operation is applied to the root node, 1 if applied to the deepest node).

- Immigrant age.

We observed how decent individuals are produced changeably as the evolution proceeds.

The ratio of crossover-produced individuals is shown in Fig.10. If the possibilities of producing decent individuals are equal for the two operations, this value would remain near 0.5 (shown as the dotted line in the graph), since reproduced individuals have no chance to become decent. Note that this graph took the fitness value as the X-axis, not the generation, so as to remove the variance of the runs. This is to show the transition of the evolution more clearly. For example, in order to express "the initial stage of evolution", the expression "the fitness of the best individual is worse than n " would be more suitable than the expression "the generation is less than n ". Since the elite strategy is adopted, the fitness value of the best individual does not get worse. Therefore, the improvement of best fitness values means the progress of generations. Fitness values are divided into a set of groups of the same width, and the average value of each group is shown in the graph.

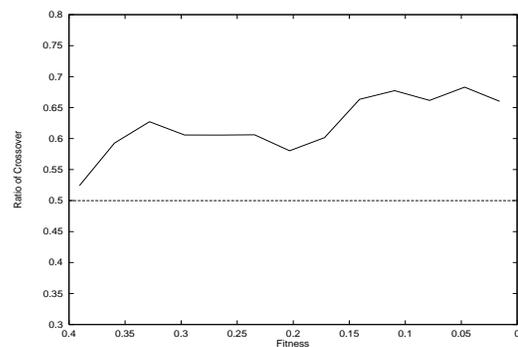


Fig. 10. Ratio of Crossover vs. Fitness

As can be seen from Fig.10, there is no significant difference in the ability of producing decent individuals in the initial stage, the crossover is clearly superior to the mutation after that initial stage. In the final stage, about 70% of decent individuals are produced by crossover.

The depth-ratio of a node to which the operation is applied is shown in Fig.11. This graph also took the grouped

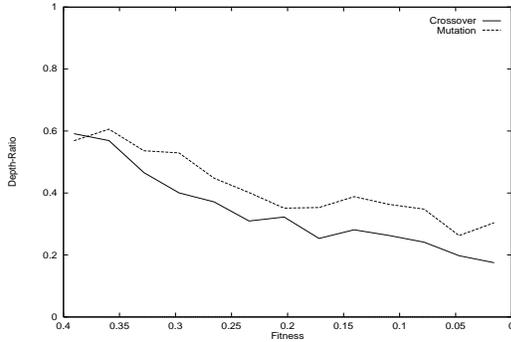


Fig. 11. Depth-Ratio vs. Fitness

fitness values as the X-axis. Note that the possibility of selecting shallow nodes was low, since only the traditional crossover was used in this experiment. From this graph, we can see that the operation at relatively deep nodes is effective in the initial stage, but the importance of applying the operation to shallower nodes increases as the evolution proceeds. Building block hypothesis can provide some good explanation to this result. The initial stage is the period at which new building blocks are built and relatively small schemata are accumulated, so that operations at deeper node are more effective. But after small schemata are produced enough, the crossover at shallower nodes is needed to accumulate them. This also explains the reason why the crossover becomes more important. Fig.11 also shows that the mutation of shallow nodes are effective as the evolution proceeds. This fact indicates that the adoption of the depth-dependent mutation may work effectively as well.

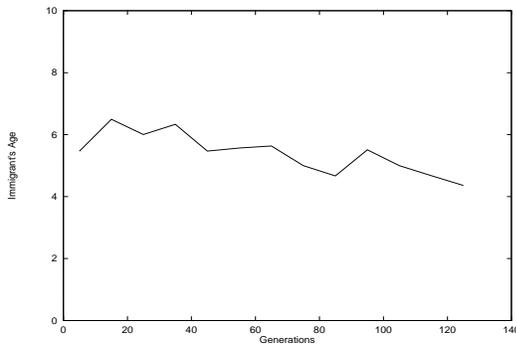


Fig. 12. Immigrant Age vs. Generation

The immigrant age of decent individuals is shown in Fig.12. Note that the X-axis is the grouped generations. It is remarkable that the average values of the immigrant age are always kept small. This graph shows that immigrants and their descendants constantly play an important role in producing decent individuals.

VI. Conclusion

This paper presented a new method in the island model GP that adopts the depth-dependent crossover to the "deme-young" individuals and the traditional crossover to others. The effectiveness of our approach was shown by the experiments, especially in several difficult problems. We also showed that the effective depth of operations changes as the evolution proceeds.

The future research will include investigating of the effects of the operation depth to improve the search ability of our method, adopting immigrant age as one of the factors in the selection, and applying our method to a more complex real-world problem.

References

- [1] John R.Koza, Genetic Programming: On the Programming of Computers by Natural Selection, MIT Press, 1992
- [2] Reiko Tanese, "Distributed Genetic Algorithms," Proc of 3rd International Conference on Genetic Algorithms, pp.434-439, 1989
- [3] T.C. Belding, "The Distributed Genetic Algorithms Revisited," Proc of 6th International Conference on Genetic Algorithms, pp.114-121, 1995
- [4] M.Tomossini, "Parallel and Distributed Evolutionary Algorithms: A Review," Evolutionary Algorithms in Engineering and Computer Science, pp.113-133, 1999
- [5] J.R.Koza and D.Andre, "Parallel Genetic Programming on a Network of Transputers," Proc. of the Workshop on Genetic Programming: From Theory to Real World Applications pp111-120, 1995
- [6] M.Miki, T.Hiroyasu, J.Yoshida and I.Ohmukai, "New Crossover Scheme for Parallel Distributed Genetic Algorithms," Proc of IASTED International Conference on Parallel and Distributed Computing and Systems, pp.145-150, 2000
- [7] J.P.Rosca and D.H.Ballard, "Causality in Genetic Programming," Proc of the Sixth International Conference on Genetic Algorithms, pp.256-263, July 15-19, 1995
- [8] Christian Igel and Kumar Chellapilla, "Investigating the Influence of Depth and Degree of Genetic Change on Fitness in Genetic Programming," Proc of the Genetic and Evolutionary Computation Conference, pp.1061-1068, July 13-17, 1999
- [9] T.Ito, H.Iba and S.Sato, "Depth-Dependent Crossover for Genetic Programming," Proc of the 1998 IEEE International Conference on Evolutionary Computation, pp.775-780, 1998
- [10] John R.Koza, "A Genetic Approach to the Truck Backer Upper Problem and the Inter-Twined Spiral Problem," Proc of IJCNN International Joint Conference on Neural Networks, pp310-318, 1992