Sparse Fitness Evaluation for Reducing User Burden in Interactive Genetic Algorithm

Joo-Young Lee and Sung-Bae Cho
Department of Computer Science, Yonsei University
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
(E-mail:[jylee,sbcho]@candy.yonsei.ac.kr)

Abstract

Interactive evolutionary computation is a technique that performs optimization based on human evaluation, and we have proposed an image retrieval method based on the emotion using interactive genetic algorithm. This approach allows to search images not only with explicitly expressed keyword but also abstract keyword such as "cheerful impression image" and "gloomy impression image". It searches the goal with a small population size and generates fewer number of generations than that of conventional genetic algorithm to reduce user's burden. But this property may derive local minimum and sometimes more poor solution than random search method owing to relatively small size population. In order to solve this problem, we suggest an idea of sparse fitness evaluation method using clustering method and fitness allocation method. This aims to allow not only to keep the advantages of interactive GA but also to improve the performance by utilizing large population.

1. Introduction

Evolutionary computation, such as genetic algorithm, evolution programming, and evolution strategies, aims to improve upon pure random search by focusing the search process on a subset of the complex and nonlinear search space. The robustness of this is strongly related with the probability that the global optimum is located in this subset. If it is not, one can only hope to find a suboptimal solution of reasonable quality. In order to circumvent this problem, many methods have been studied and proposed.

Interactive Evolutionary Computation is a technique that performs optimization based on human evaluation [1]. A human operator can obtain what he/she wants through repeated interaction with computer. It has a special advantage, which is to adopt user's choice as fitness, when fitness function cannot be explicitly defined. This property allows to develop an effective human-oriented system.

We have proposed an image retrieval method based on the emotion using interactive genetic algorithm [2]. This approach allows to search images not only with explicitly expressed keyword but also abstract keyword such as "cheerful impression image" and "gloomy impression image". However, interactive genetic algorithm searches the goal with a small population size and generates fewer number of generations than that of conventional genetic

algorithm to reduce user's burden. This property may derive local minimum and sometimes more poor solution than random search method owing to relatively small size population.

In order to solve this problem, we suggest an idea of sparse fitness evaluation using clustering method and fitness allocation method. This method aims to allow not only to obtain the advantages of interactive GA but also to improve the performance by using large population.

In this paper, we introduce image retrieval system using interactive genetic algorithm briefly and suggest a method to reduce user burden while maintaining a large number of individuals.

2. Related Works

2.1. Interactive Genetic Algorithm

Genetic algorithm is a model of machine learning derived by the procedure of evolution in nature [3]. This is performed by creating a population of individuals that are represented by chromosome. The chromosome is the string that can be thought of as the human gene. The individuals in the population go through the evolution. This procedure takes the evolutionary procedure in which different individuals compete for resource in the environment. Some of better individuals are more likely to survive, and propagate their genetic material to offsprings. The procedure of a simple genetic algorithm is as follows.

```
t = 0;
InitializePopulation P(t);
Evaluate P(t);
While not done do
    t = t + 1;
    P' = SelectParents P(t);
    Recombine P'(t);
    Mutate P'(t);
    Evaluate P'(t);
    P = Survive P, P'(t);
End While
```

The algorithm starts with initial population. In the beginning, the fitness value of each individual is evaluated to determine how appropriate the individual is for the given

problem. In general, individuals in an initial population are randomly generated. Next, two individuals that have relatively high fitness value are selected from the population. Selected individuals can be regarded as parents. New individuals called children are created by recombinating the chromosomes of parents. Here, selection and crossover operators are used. Mutation replaces existing gene code with randomly generated code. It performs a function that puts potentially good genes or properties that may be lost during applying genetic operation into the population.

Interactive genetic algorithm adopts user's choice as fitness, when fitness function cannot be exactly determined. This property allows developing system based on human intuition or emotion. It has not been studied broadly so far, but a property which user determines fitness allows to adapt interactive genetic algorithm in the field of graphics or art. For example, Caldwell and Johnston applied it to tracking criminal suspect [4]. This system produces montages by the fitness based on the face of criminal suspect by witness's account. Baker implemented line drawing system based on the user's aesthetic criteria using interactive genetic algorithm [5]. The criteria of this system are attraction, amusement, control and so on. However, it was difficult to make specific shape to want exactly, because the size of population was small. To solve the problem, he tried to expand search space by increasing mutation rate.

2.2. Clustering Techniques

Clustering refers to the process of grouping samples so that the samples are similar within group [6]. The groups are called clusters. This technique is divided into two categories. One is agglomerative clustering and the other is partitional clustering.

The general agglomerative clustering algorithm is straightforward to describe. If the total number of samples is n, it begins with n clusters, each consisting of one sample, repeats following step for a total of n-1 times, finds the most similar clusters C_i and C_j, and merges C_i and C_j into one cluster.

Different hierarchical clustering algorithms are obtained by using different methods to determine the similarity of clusters. One way to measure the similarity between clusters is to define a function that measures distance between clusters. This distance function typically is induced by an underlying function that measures the distance between pairs of samples. In cluster analysis as nearest neighbor techniques, the most popular distance measures are Euclidean distance and city block distance. Single-linkage algorithm, complete-linkage algorithm, average-linkage algorithm, and Ward's algorithm belong to this method [7].

Agglomerative clustering creates a series of nested clusters. This contrasts with partitional clustering in which the goal is usually to create one set of clusters that partitions the data into similar groups. Samples close to one another are assumed to be similar and the goal of the partitional

clustering algorithm is to group data that are close together. In many of the partitional algorithms, the number of clusters to be constructed is specified in advance. Forgy's algorithm, k-means algorithm, and isodata algorithm belong to this category [8].

Clustering techniques allow the division into subgroups to be done automatically, without any preconception about what kinds of groupings should be found. Cluster analysis has been applied in many fields. In image analysis, especially, clustering can be used to find the groups of pixels with similar gray levels, colors, or local texture, in order to discover the various regions in the image.

3. Image Retrieval System using Interactive GA

3.1. Overview

The entire system is constructed as shown in Fig. 1. In preprocessing step, we perform wavelet transform for every image in the database and store the average color, and the indices and signs of the m magnitude wavelet coefficients in a search table.
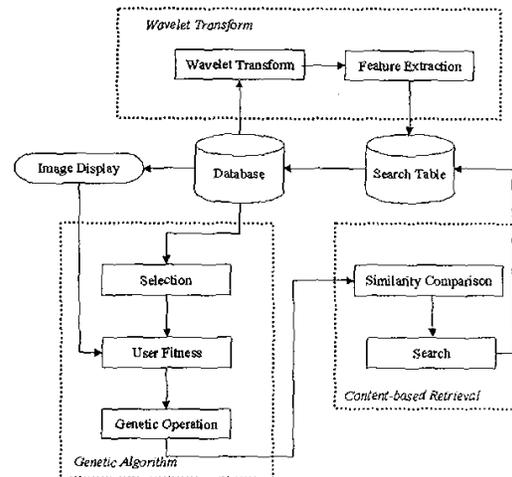

Fig. 1   The system structure.

Interactive genetic algorithm is used to search images. A system displays twelve images, obtains the fitness values from human, and selects candidates based on the fitness. Here, a chromosome is represented by an array that consists of index of wavelet coefficients. The strategy of selection is governed by expected frequency of each individual, and we use one point crossover that proceeds by selecting a point and swapping a part of chromosome on the basis of it. In that case, two among individuals having high fitness are mated at random. In detail, a horizontal and a vertical crossover are used. Those allow to exchange color and shape features respectively. Mutation is not adopted.

To find better twelve images, the stored image

information is evaluated by each criterion. Twelve images of the higher magnitude value are provided as a result of the search. At this time, the similarity between potential target image and candidate image is calculated by the following equation.

$$\| Q,T \| = w_{0,0} \,|\, Q[0,0] - T[0,0] \,| + \sum_{i,j} w_{i,j} \,|\, Q[i,j] - T[i,j] \,| \qquad (1)$$

$Q[i,j]$ and $T[i,j]$ represent single color channels of wavelet decomposition of the query and target images, and $Q[0,0]$ and $T[0,0]$ mean overall average intensities of those color channels. The system repeats this process to search new candidates until user finds the image that he wants.

Fig. 2   The system interface.

3.2. Performance

The system is programmed in Microsoft Visual C++ on Pentium PC. The size of image database is 2000. In order to search more efficiently and quickly, a searching table is constructed by a batch job over the 256×256 JPEG images. It maintains signs and index of wavelet coefficients. The crossover rate is 0.6.

Initial population consists of twelve images selected randomly. As user gives images the fitness based on the similarity with what he/she wants, the system presents new images in the next generation using genetic algorithm. Fig 2 shows the user interface of the system. This procedure is repeated until the user obtains the image that is most similar to what he has in mind. In case the results of the next generation are not satisfiable, the system allows user to go back to the previous generation.

It is pretty hard to measure the convergence in the case of interactive genetic algorithm. Toward this goal we have attempted to show the change of fitness according to the generations and compare the results obtained. We conduct GA convergence experiment with ten subjects and Fig. 3 shows the average curves of the average and best fitnesses for ten generations in the case of searching the image of

gloomy impression. We can see that the fitness is effectively increased by adopting user's evaluation, despite the searching space of chromosome encoded using wavelet coefficients is very large. Fig. 4 shows the result images converged. There are more images of gloomy impression in the 8th generation than those in the beginning.
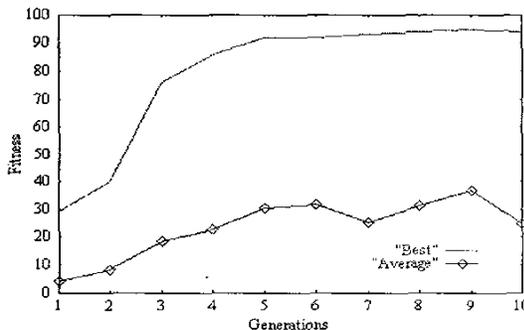

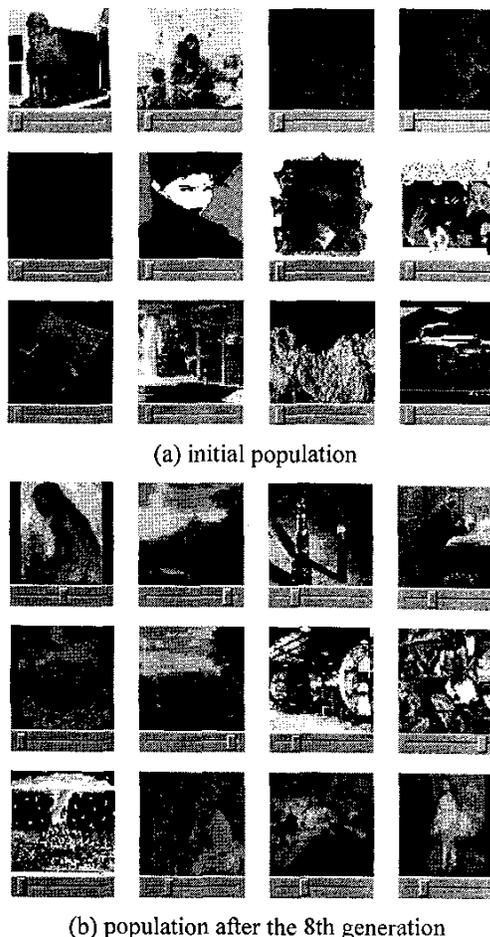Fig. 3   The best and the average fitnesses in the case of searching the image of gloomy impression.

(a) initial population

(b) population after the 8th generation

Fig. 4   Comparison of the results in case of the task of gloomy impression.

4. Sparse Fitness Evaluation Method

Interactive GA has to search the goal with a small population size and generate fewer number of generations than that of normal genetic algorithm. Although it is for reducing user's fatigue, it may derive local search caused by partial evaluation. In order to avoid this problem, we suggest a sparse fitness evaluation method using clustering method and proper fitness allocation method. This method allows a system using interactive GA to evaluate many and various individuals by using a large population.

The basic idea is to perform the evaluation by two step mechanism. We divide all individuals in the population into subgroups by clustering method, evaluate representatives of each subgroup and allocate fitnesses to the remaining individuals of each subgroup. Fig 5 and 6 show the simple algorithm and basic idea.
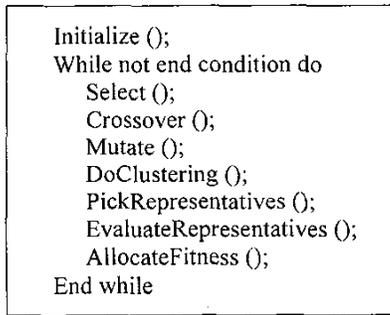
```
Initialize ();
While not end condition do
    Select ();
    Crossover ();
    Mutate ();
    DoClustering ();
    PickRepresentatives ();
    EvaluateRepresentatives ();
    AllocateFitness ();
End while
```

Fig.5   Simple algorithm.
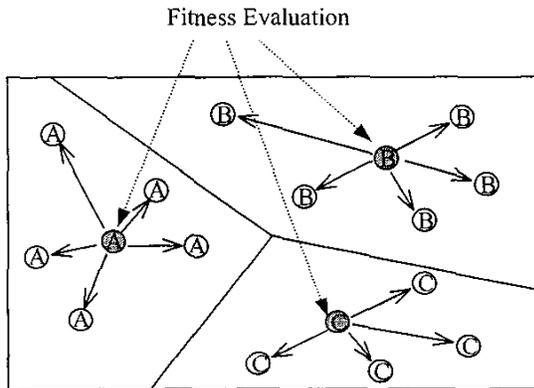
Fitness Evaluation

Fig. 6   Sparse fitness evaluation.

Here, we can consider two issues: which clustering method is used and how to allocate fitnesses to the remaining individuals. In this paper, we test the sparse fitness evaluation method using k-means algorithm and fitness allocation by distance and evaluate the performance of proposed method with DeJong's five functions.

4.1. Clustering Method

In order to divide population into subgroup, we can use k-means algorithm to construct clusters as subgroups. It is one of the simplest partitional clustering algorithm. Besides the data, input to the algorithm consists of k, the number of clusters to be constructed, samples called seed point. The seed points could be chosen randomly, or some knowledge of the desired cluster structure could be used to guide their selection. Fig. 7 shows simple k-means algorithm. This algorithm recomputes the centroids of clusters as soon as a sample joins a cluster and makes only two passes through the data set. In this paper, we select 10 seed points $k = 10$ randomly, and use Euclidean distance to calculate distance between two individuals.
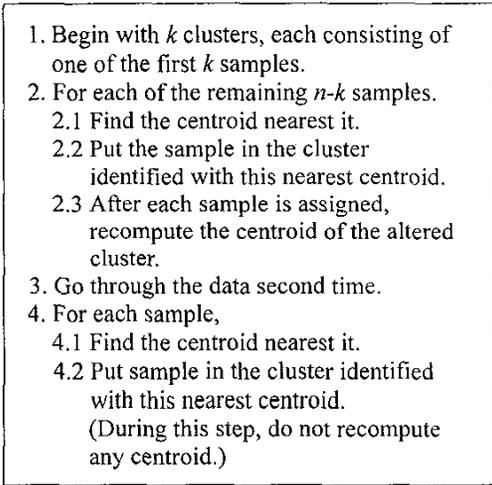
```
1. Begin with k clusters, each consisting of
   one of the first k samples.
2. For each of the remaining n-k samples.
   2.1 Find the centroid nearest it.
   2.2 Put the sample in the cluster
       identified with this nearest centroid.
   2.3 After each sample is assigned,
       recompute the centroid of the altered
       cluster.
3. Go through the data second time.
4. For each sample,
   4.1 Find the centroid nearest it.
   4.2 Put sample in the cluster identified
       with this nearest centroid.
       (During this step, do not recompute
       any centroid.)
```

Fig. 7 K-means algorithm

4.2. Fitness Allocation Method

There can be various methods to allocate fitness to individuals. We take one of the simplest and intuited methods. That is a fitness allocation by a distance within cluster. We elect an individual located in the center as a representative of the subgroup. The representative is evaluated by objective function and fitnesses of the rest of individuals are allocated based on obtained fitness of the representative. In this time, we give an individual fitness in proportion to a distance from the representative. This distance is calculated following equation.

$$dist(x) = \sqrt{\sum_{i=1}^{n} (c_i - x_i)^2} \qquad (2)$$

5. Experiments

For our experiments, we use the DeJong's five functions and evaluate a performance between general GA and proposed method. The performance measure is GA convergence characteristics versus computation time. We compare three methods. One is proposed method having population size of 100 and the number of cluster of 10. Therefore, 10 representatives are only evaluated by objective function and the rest 90 individuals are given fitnesses by fitness allocation method, another is a conventional method having 100 population, and the rest is a conventional method having 10 population. Table 1 shows GAs used in our experiment. During our experiments, we have used the parameter settings presented in Table 2. The test results are shown in Fig 8.

In the experimental result, we can see that the performances of conventional method with 100 population size and the proposed method are nearly the same. However, when the proposed method is compared with the conventional method having a population size of 10, the performance of the proposed method is better than that of conventional in case of all DeJong functions. Therefore, we can say that the proposed method can be a solution to some special problems, which are applied interactive genetic algorithm.

Table 1. GAs used in the experiment.

| P_1 | Proposed method using k-means algorithm for clustering and fitness allocation by distance |
| N_1 | conventional method having 100 population |
| N_2 | conventional method having 10 population. |

Table 2. Parameters of genetic algorithm.

| Parameters	Value
Population	P_1 (100), N_1 (100), N_2(10)
Crossover rate	0.9
Mutation rate	0.001
# of cluster	10
# of generation	200

6. Concluding Remarks

This paper has proposed an approach that searches an image with human preference and emotion using genetic algorithm. We use wavelet transform to extract image features and interactive genetic algorithm to search the image that user wants. When user gives appropriate fitness to what he or she wants, system provides images selected based on the user's evaluation. This approach allows to search not only explicitly expressed image, but also abstract image such as "cheerful impression image", "gloomy impression image", and so on.

However, interactive genetic algorithm has to search the

goal with a small population size and generate fewer number of generations than that of conventional genetic algorithm to reduce user's burden. Therefore we have proposed an idea that is a sparse fitness evaluation method using clustering method and proper fitness allocation method. It allows not only to obtain the advantages of interactive GA but also to improve the performance by using a large population.
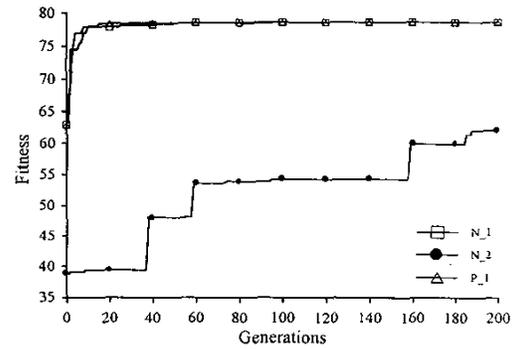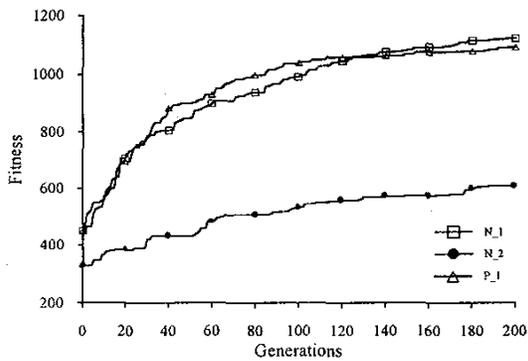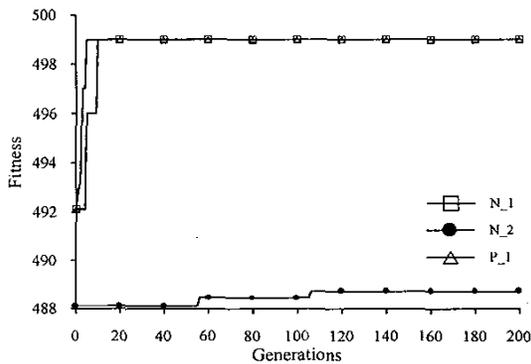

(a) DeJong's function 1
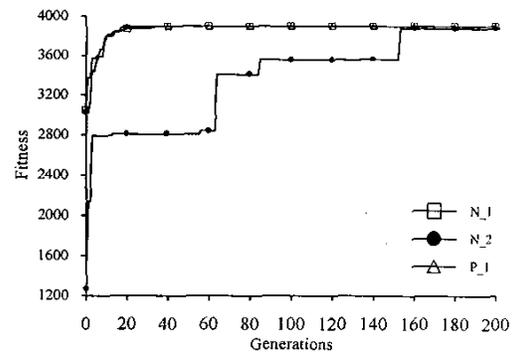
(b) DeJong's function 2
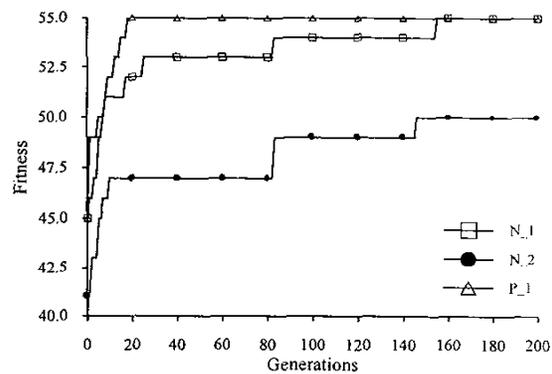
(c) DeJong's function 3

(d) De jong's function 4

(e) DeJong's function 5

Fig. 8   Comparison results of conventional methods having a population size of 100 and 10, and the proposed method.

7. References

[1] H. Takagi, "Interactive evolutionary computation: Cooperation of computational intelligence and human kansei," *Proc. of 5th Int'l Conf. on Soft Computing*, pp.41-50, Iizuka, Japan, 1998.

[2] J.-Y. Lee, and S.-B. Cho, "Interactive genetic algorithm for content-based image retrieval," *Proc. of Asian Fuzzy Systems Symposium (AFSS'98)*, pp.479-484, Masan, Korea, June 1998.

[3] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company Inc. 1989

[4] C. Caldwell, and V.S. Johnston, "Tracking a criminal suspect through "face-space" with a genetic algorithm," *Proc. of 4th Int'l Conf. on Genetic Algorithms*, pp.416-421, 1991.

[5] E. Baker and M. Seltzer, "Evolving line drawings," *Proc. of Graphics Interface'94*, pp.91-100, 1994.

[6] E. Gose, R. Johnsonbaugh, and S. Jost, *Pattern Recognition and Image Analysis*, Prentice Hall PTR, 1996.

[7] C.H. Chen, *Statistical Pattern Recognition*, Hayden, Washington, D.C., 1973.

[8] K. Fukunaka, *Introduction to Statistical Pattern Recognition*, Academic Press, New York 1990.