A Survey on Accelerating Evolutionary Computation Approaches

Pei, Yan
Graduate School of Design, Kyushu University

Takagi, Hideyuki
Faculty of Design, Kyushu University, Kyushu University

出版情報：International Conference of Soft Computing and Pattern Recognition. 2011, pp.201-206, 2011-10-14. IEEE
バージョン：
権利関係：

A Survey on Accelerating Evolutionary Computation Approaches

Yan Pei
Graduate School of Design, Kyushu University
Fukuoka, Japan
peiyan@ieee.org

Hideyuki Takagi
Faculty of Design, Kyushu University
Fukuoka, Japan
http://www.design.kyushu-u.ac.jp/ takagi

***Abstract*—In this paper, we review the research on acceleration convergence approaches of evolutionary computation (EC) and its concrete application in the academy and industry. Evolutionary computation uses iterative progress, which is often inspired by biological mechanisms of evolution, to solve the problems that are multi-modal, multi-objective, discontinuous, non-differential, noisy and not well-defined. In this survey, many acceleration approaches are summarized and clustered in recent two decades. Applications of the acceleration approaches are included. We propose three promising research directions and their concrete approaches. These include including search space landscape approximation, search space projection and search strategy study, and comprise the main further research directions to be implemented an efficient EC search. Finally, we discuss the future research on accelerating convergence approaches of EC, and motivate some new approaches.**

***Keywords*-evolutionary computation; acceleration convergence; search space landscape; projection space; search strategy**

I. Introduction

Evolutionary computation (EC) comprises bionic optimization algorithms. EC simulates the production and evolution process of all the lives and intelligence agents, which have developed not only on the basis of Darwin's principles of natural selection and *survival of the fittest*, but also have utilized the theory of genetic and mutation by Gregory Mendel to maintain an optimized result, meanwhile attempting to find a better solution. EC is a probabilistic search optimization technique, which uses computational models of evolutionary processes as key elements in the design and implementation of computer-based problem solving systems [1]. There have been four well-defined evolutionary computation algorithms (ECs), which have served as the basis for the most of activities in the field of evolutionary computations: genetic algorithms (GA) [2] [3], evolution strategies [4] [5] [6], genetic programming [7] [8], and evolutionary programming [9].

Algorithm design is a core issue to EC, and it is also an important aspect of EC research. Although great success has been achieved in academic research and industrial applications, EC has also encountered many challenges. There are two important research topics. One is extending the application domain, which requires an efficient implementation strategy for problems that conventional approaches cannot solve or efficiently solve. The other is making efficient algorithms, which improves the existent EC to let them be more efficient or convergent to some states. It is therefore important to design efficient algorithms to accelerate the EC convergence.

To avoid premature is a key aspect of the EC convergence. In the EC search process, all the objective solutions are made after coding. However, the individuals' creation process is based on randomization, when the distribution of individuals is very ”massive” in the search space, after the initial EC operations, premature may happen.

Another issue is the individual diversity that influences the EC convergence performance. We hope the individuals are created near the global optimum, which might be obtained after the EC operation. But since the individuals' creation is a randomization process, we cannot always obtain the individuals near the global optimum. We therefore need to establish a search space description approach to direct creation of individuals and search process, which must be an efficient way to accelerate EC convergence.

The remainder of this paper is structured as follows. First the state of the art of evolutionary computation accelerating approaches and applications in the last two decades are summarized (Section II). For the further research on EC acceleration approaches, the main finding is proposed and an outlook on the future work is given (Section III). Finally, we discuss further research directions (Section IV) and conclude the whole paper (Section V).

II. Evolutionary Computation Acceleration Approaches

A. Coding Technology

The coding technology is a basic issue for EC algorithm design, and different coding methods influence the algorithm performance and concrete applications. For GA, for example, many coding technologies have been proposed. These include messy GA by Golderg [33], Delta-coding GA by Mathias [34], dynamic parameter encoding GA by Shcraudolph et al [35] and real coded GA [36]. These coding technology develops the EC search space and make a basement for a variety of search strategies in the respective space. This influences the EC performance.

B. Population Constitution

1) Hybrid Population Resultant Approaches: It is an important EC acceleration technology to construct a new population from one generation to the next. In this process, there are two key aspects to be considered. One is to keep the individuals' diversity, which are distributed in the whole search space averagely. The other is to let the individual as much as possible near the global optimum point, which makes it possible to obtain the global optimum.

Reference [10] proposed a virtual population technology to reconstruct a new population to achieve the key aspects mentioned above. A candidate's population is a population of candidate solutions made up of the current population and three populations derived from it. There are some important aspects of the candidate's population. Firstly, it provides much more diversified individuals for selection in the resulting population, which means solution space will be searched more thoroughly. Secondly, the better individuals from virtual population are always retained. Reference [10] shows that a fraction of individuals are accelerated by a gradient method to further enhance the convergence speed, so that the value of the fraction can easily be found by experimentation.

There is another hybrid population resultant technology using elitist selection and stochastic universal sampling approaches [11]. It separates the population into two groups, one coming from stochastic universal sampling and the other coming from elitist selection. It conducts the crossover operation within each group and the mutation operation between the two groups. Stochastic universal sampling prefers fitter individuals but also gives a chance to worse individuals. With elitist selection, only the fittest individuals are taken, which effectively creates a suitable compromise between fast convergence and avoidance of a loss of diversity.

Pipeline based hyper-population resultant was proposed in reference [12]. The conventional GA and its variants, including crossover, mutation, and selection operation are all performed on the individuals of the same generation in which the population size of each generation is fixed. In pipeline based hyper-population resultant, the EC operations can be performed, while offspring candidates are not generated and their fitness values are not calculated completely. When an offspring is generated in the offspring pool and some unprocessed parent chromosomes still remain in the base generation, the pipeline based hyper-population resultant allows the generated offspring to mate with one of the unprocessed chromosomes immediately. Thus, the long waiting time in each generation does not limit the evolution speed.

2) Dynamic Fitness Threshold: Fitness threshold gives higher priority to the fitter individual. It is initialized by the user, and only if the new individual is greater than the fitness threshold one, the new individual is put into the population, otherwise this randomly generated individual is not considered. The fitness threshold will tune from the previous generation. The policy improves the fitness values, which will always be increasing from one generation to next generation. When the population lacks diversity, a local optimum may happen. Research on how to use the dynamic fit threshold policy and avoid the local optimum could therefore be promising.

3) Fitness Scaling Technology: Fitness scaling technology can be used to tune the individual fitness. It uses some linear or non-linear transformation to avoid the premature in the selection operation, but this handling destroys the evaluation rule. For interactive EC (IEC) application, it may be a good approach to assist human's evaluation, which is sometimes inconsistent.

Fitness scaling technology assumes that the best solution in the current population is closest to the global optimum. If this assumption is true, then searching the solution space in this neighborhood will produce solutions closer to the global optimum. Conversely, if the assumption is false, this approach will lead to a local optimum. We will therefore search the opposite direction to avoid the problem (local optimum), and search beyond the best solution in the current population to find the global optimum and escape the local optimum. References [10] and [13] use an approach to create population which conducts local search near the best solution to find a global optimum and avoids the local optimum to solve load flow problem in the power system.

C. EC Operation Alteration Technology

1) Engineered Conditioning Operator Approaches: Engineered conditioning operator [14] [15] [16] is the same conventional optimization search strategy, i.e. *moving to the best adjacent point in the search space*. Engineered conditioning uses the dominant individuals of the current population in the search space and compares their strength. Three local search are performed, the superset test, the substitute test, and the subset evaluation test. Those tests are looking for better individuals of greater cardinality, equal cardinality, and lesser cardinality.

Reference [15] proposed the engineered conditioning operator based GA to accelerate convergence on the multiple fault diagnosis application. Reference [16] extends the research work of reference [15], and calls it the engineered conditioning operator as local improvement operators [17] that is domain-specific. In its improvement, a more substitute test is added in the engineered conditioning operator, called E2C based GA to accelerate the convergence.

2) Age Conception Evolution based Directed Searching Approaches: This approach [18] is based on the two conceptions in the evolution process; one is the age, the other is direction. The age shows the diversity feature of the population, and the direction shows whether the next evolution is forwards to the optimum regions.

The advantage of this approach is improvement of the convergence without decreasing the diversity among the individuals. One is a direction operator which determines the direction of the search according to the fitness score. One is a zero-mean Gaussian operation, which is used for a perturbation and adds to a parent in order to generate an offspring. Reference [18] uses this approach to solve the seven parameters friction model optimization problem in system cybernetics.

It is an elegant approach to the IEC fitness evaluation problem. Because the IEC fitness value comes from the human's subjective evaluation, there is no absolute scale to judge the solution, which can lead to incongruence between the parent and its offsprings. It can use the conception of age and direction to adjust the fitness value (human's subjective evaluation) to construct a *directed based human's evaluation model* to improve the human's subjective evaluation and reduce the fatigue.

3) Elitist Model: EC use the elitist model [11] to preserve *the best fitness individual obtained so far* within a population. Reference [19] proposed two methods to keep the best fitness individual,. One makes two copies of the best fitness individual in the old pool, and places it in the new pool. The other compares the child and its parent, the better one survives.

In recent EC research, most of the EC applications use the elitist model policy to implement the algorithms. Some EC convergence theory is based on the elitist model, so for the EC application and convergence theory without the elitist model, it will be a great and challenging topic in the EC research domain.

In single objective GA (SGA), elitism is an operator that ensures that the best chromosome found so far exists in iteration. This can be realized by simply copying the best individual to the next generation. For multi-objective optimization, the elitism strategy no longer remains trivial, as there is no longer a single best design to copy. Because of this, various elitist strategies have been proposed [20]. A simple elitist approach [21] is that parents compete with offspring, in that the child population is combined with the original parent population, and the entire group of individuals is ranked. The best of these individuals are retained as the parents for the next generation.

D. Hybrid with Other Approaches

1) Local Search Algorithm: Local search algorithms move from solution to solution in the space of candidate solutions in the search space, until a solution deemed optimal is found or a time bound is elapsed. A local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution, but it is only possible if a neighborhood relation is defined on the search space. Adding a randomization step to a gradient search algorithm can improve its performance, but for many large and complex search spaces this may not be enough. Many experiments have shown that such randomization has little effect. For hybrid EC with a local search algorithm issue, a different neighborhood relation strategy makes up different accelerated EC approaches [22] [31].

Reference [30] proposed local search and global search with a surrogate model to accelerate the ECs. Reference [32] used a uni-modal function to approximate search space and add only the peak point of the uni-modal function as a powerful individual to accelerate the GA. As only one of many individuals is replaced with the powerful individual, it does not destroy the original EC search (low risk) and it becomes a really powerful parent when it is close to the global point (high return). The experiment [32] shows a good performance for this low risk and high return approach.

2) Constraint Satisfaction: Constraint satisfaction is the process of finding a solution to a set of constraints that impose conditions that the variables must satisfy. A solution is therefore a vector of variables that satisfies all constraints. The techniques used in constraint satisfaction depend on the kind of constraints being considered. Constraint satisfaction based ECs [13], most of them are incomplete in general. That is, they may prove it unsatisfied, but not always.

This approach uses some constraint rule, which is based on specialty domain knowledge to modify the unsatisfied individuals after crossover and mutation operation, i.e., it conducts the local search near the unsatisfied individual space to check whether the optimum exists.

3) Simulated Annealing: Simulated annealing (SA) is a generic probabilistic meta heuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global optimum of a given function in a large search space. There is the ability to escape the local optimum by incorporating a probability function in accepting or rejecting a new solution, and a cooling schedule has been used to accelerate convergence. It is often used when the search space is discrete.

Reference [23] proposed a simulated annealing method, which is used to accelerate the convergence of GA by applying the simulated annealing test for all the population members. The SA test allows the acceptance of any individuals at the initial steps in the search, but only the good individuals have the priority to be accepted as the generation increases. The experiment test shows that the SA method is efficient for helping GA to escape from the local optimum to prevent the premature convergence.

4) Artificial Neuron Network: An artificial neural network (ANN) is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data [39].

For the recent research, there are two topics for ANN based ECs. Firstly, EC uses ANN as a direction tool to find the best search direction, which uses landscapes information of the EC search space to train the ANN and with this information the EC search direction is decided [24]. Secondly, EC uses the trained ANN, which confirms an input-output relationship for special application with domain knowledge. Normally, three-layer structure ANN, which has an input layer, a hidden layer and an output layer, is considered to be used in the applications. Multilayer feed-forward neural networks with error back-propagation algorithm have been applied to accelerate the execution of GA loop [25].

E. Stop Criteria

An important but always ignored problem that is to design proper stop criteria for EC. The EC convergence concept is a good tool for designing the stop criteria, such as Markoff chain, fixed point theory, etc. However, the convergence concept is time unlimited in describing the system's behavior. For some complex problem, we cannot use it to design the stop criteria. There are several method proposed for designing the stop criteria [26] [27] [28].

1) Fitness boundary value method: When the individual's fitness value is more than a certain boundary value, EC stops.

2) Time boundary value method: When the algorithm running total time is more than a certain boundary value, EC stops.

3) Individual or generation number boundary value method: When the individual total numbers or generation total numbers is more than a certain boundary value, EC stops.

4) Fitness number boundary value method: When the different fitness numbers that the ECs search generates is more than a certain boundary value, EC stops.

In the ECs search process, there may be the same individuals to be created, i.e. their fitness value are the same. It is not enough to judge the searched space quantum by the individual numbers, however, it is guaranteed when searching the enough different fitness number, which shows most of the search space has be searched. This approach can improve the individual number and generation number boundary value method to design new stop criteria.

5) Fitness increasing probability boundary value method: When there are n number of individuals' fitness value that have not increased, the $n+1$ individual can obtain the increasing probability as $p = 1/(n+1)$. It can define when $p < e$, EC stops, i.e., when $1/e$ number of individuals whose fitness value have not increased, EC stops.

6) Online and offline performance: The performance is based on static relations applied during the execution of the algorithm. There are two types of performances [29]. Online performance (OnLP) is defined to reflect the character of average fitness of the whole individuals. During execution of SGA, online performance converges to a stable value, i.e. stationary state, and the solutions formed by the algorithm become stable. The off-line performance (OffLP) is similar to the OnLP, but it gives a lot of importance to the best fitness value. OffLP converges to a stable value while the number of evaluations is increasing, and the probability of finding a better solution is decreasing quickly. Therefore, the optimization process can be stopped to avoid any waste of time.

F. Mutation and Crossover Rate Chosen Problem

A key topic on EC operations is the rate chosen problem. Mutation adjusts the diversity of the population and partially decides the convergence of search process. If the mutation rate is small, but the convergence speed is high, the diversity becomes worse, and the search points often converge to a local optimum and lead to the premature. However, mutation with a high rate may result in the loss of the better individuals.

Another major genetic operator is crossover, which is designed to generated offspring in the hope that better fitness is achieved through exchanging partial genetic information of two parents. A valuable research topic is how to balance the EC convergence and performance by tuning the EC operations rate, and design the adaptive EC operations rate or find more efficient operations [31].

III. Further Acceleration Approach Research

We have given an overview of the accelerating EC convergence approaches in the last two decades in Section II, and described the related papers of EC acceleration approaches applications. In this section, we will present a brief proposal for further research on the EC acceleration approaches.

EC operations do not directly use the landscape information of search space. If we can use more landscape information directly, EC convergence may be accelerated. For this idea, there are three concrete approaches. The first is the approximation of landscape with a simpler shape to find new elite in the new search space. The second is to project search space to other dimensional search space to obtain easy search information to find the global optimum in the projected space. The third is that we may conduct efficient search strategies in original space and projected space, it can help accelerating or observing the EC convergence, and adjusting different search strategies adaptively. The three concrete approaches on accelerating EC convergence are the main works in our further research.

A. Approximation of EC Landscape

If we can reduce complexity of the searching space, it can become easier to reach to the global optimum in the approximation search space. It is not the real global optimum but may be a neighbor around the global optimum. From this viewpoint, we can use the gradient search, local search or some related search algorithms to find the global optimum in the approximation search space. Therefore, it is easy to reach to the real global optimum from the neighbor point. For the concrete approximation landscape methods, there are so many mathematical approaches or computation-based that can be used.

1) Space Frequency Information: We can use Fourier or Wavelet transform to find the global optimum area with frequency landscape information.

2) Curves: We can estimate search space peaks from search points using Spline function, Bezier curve, and other curve lines.

3) Signal Processing Filtering: We can assume fitness fluctuation due to the complex landscape as noise. Then a noise reduction filter may approximate the complex landscape with a simpler shape.

4) Differential Information: It is an efficient method to use gradient information by difference vectors among searching points to obtain search direction information.

5) Polygons: Approximation landscape with small triangles is an approximation method, as finite element method and polygons are used in computer graphics.

6) Clustering: We can separate complex landscape into multiple simpler landscapes by clustering methods.

B. Search Space Projection to Other Dimensional Space

If we can use some projection algorithms to project EC individuals from its original space to another higher or lower spaces, and conduct some search strategies to search, it should be efficient and EC convergence may be accelerated.

1) To higher dimension: Supposing original search space is n-D space and projection space is k-D space ($k > n$), support vector machine projects data onto a higher dimensional space and finds linear separation in the k-D space for the nonlinear separation of clusters in the n-D space. If we can apply the same way of thinking to global optimization, we can reach to the global optimum in the k-D space quickly, and then search to the global optimum in the n-D space from the neighbor point.

2) To lower dimension: Supposing we project search points, which are in a n-D, onto m-D space ($m < n$), it reduces dimensional complexity though some pieces of information are gone. In the lower dimensional search space, we search the global point in a simpler space, and then search the real global optimum in its original search space. This is the same approach mentioned in the above.

Reference [40] proposed an obtaining elite approach in projected one-dimension search space by this strategy, and the experimental results show the better acceleration of EC convergence, especially in the initial generations.

C. Search Strategy Study

When we transfer the search space from A to B ($SpaceA$ to $SpaceB$), there are several searching or observing strategies, which can provide the information for the algorithms execution. Suppose the original search space is $SpaceA$ (the individuals are all in $SpaceA$), and we use some mapping methods to let the $SpaceA$'s individuals to another space, and we call it $SpaceB$. In $SpaceB$, we create the new individuals (the neighbors of the old ones), and then three strategies are possible.

1) Strategy A: Strategy A is to map the new neighbors individuals back to $SpaceA$ as a new generation, then to search in $SpaceA$, and then to map to $SpaceB$ again to find new individuals, looping this policy until the global optimum is found.

2) Strategy B: Strategy B is to keep search in space B, and when the some fitness values meet a certain condition, we map the individuals back to $SpaceA$ again to judge how to deal with them further, then stop or map to $SpaceB$ to continue to search.

3) Strategy C: Strategy C is to search the global optimum in the $SpaceA$ as the similar to conventional EC search, but to monitor the search situation of the $SpaceA$ in $SpaceB$ and feedback search control to the search in the $SpaceA$.

The main research point of the search strategy is to find or design the projection that helps EC search. The choice for strategy A, B, or C will be decided after analyzing the characteristics of the projection.

D. EC Fusion with Other Soft-computing Approach

Conventional computational approaches could model and precisely analyze only relatively simple systems. More complex systems arising in biology, medicine, humanities, management sciences, and similar fields often remained intractable to conventional mathematical and analytical methods. So in the early 1990s, a so-called soft computing [37] technology was proposed, which is includes evolutionary computation, neural network (NN), fuzzy system (FS) and other computation intelligence technologies. Ever since their proposal, fusion of these technologies has been an active research direction [38] [39], such as EC+NN, NN+FS, and so on.

Soft computing deals with imprecision, uncertainty, partial truth, and approximation to achieve tractability, robustness and low solution cost. For the different technique in soft computing, it has its unique features. It is worth investigating how to use one technique's advantage to compensate the other technique's deficiency. For an instance, neural networks has the memorial function, and the genetic algorithm is a best search optimization tool. It might be good to fuse both techniques in such a way that the memorial function of neural networks is used to direct the search direction of genetic algorithms so that the genetic algorithm's convergence can be accelerated [24].

IV. Discussion

There are two main research directions of EC. One concerns extending the EC application domain, and the other is making evolutionary computation algorithms more efficient. For the EC acceleration approaches issue, the EC search strategy is not more efficient than the conventional approaches, except the problem is only solved by EC.

There are several reasons to influence the EC performance. The EC search process focuses on the simulation of the population evolution, and ignores the environment's influence to the population. Re-sampling phenomena happen when EC search, which is the randomization process. So many ECs use the elitist strategy to increase the computational complexity. For solving those issues, we can establish the ECs model considering the environment's effect, reduce the EC re-sampling in the search process, and consider a more efficient strategy rather than elitist strategy to accelerate the EC convergence. These concern the concrete approaches to accelerate EC convergence.

Another research direction is to compare the EC algorithm with the conventional optimization algorithm. From the conventional optimization algorithm search strategy, we could obtain inspired ideas to accelerate EC, and compose EC with another conventional optimization algorithm as a hybrid one to investigate how to accelerate EC.

It could be worthwhile to study the relationship between the EC acceleration approach and EC convergence theory. In the last two decades, many accelerated EC approaches were proposed, but there was few theoretical explanation to discover the principles of why and how accelerated EC approach works. It is known that the scientific research history reflects a process of the reactions between the approach and theory. Reference [2] proposed a schema theorem that has proved to be correct until now, no perfect theory has been proposed yet to support the EC related domains' development. Further work on the phenomena in many accelerated EC approaches is necessary to develop the theory that can support and guide the further development of EC. This way EC acceleration research can obtain better results towards the final objective.

V. Conclusion

In this paper, we gave an overview on the concrete acceleration EC convergence approaches. Both of the acceleration approaches and their applications are included from the last two decades. We propose three promising approaches to accelerate EC convergence in this research direction. These concern the approximation of EC landscape, search in a different dimension space and search strategy study. Finally, we present remaining problems in the acceleration

EC convergence research, and give an outlook for the further work on this topic.

Acknowledgment

Yan Pei would like to thank Yoshida Scholarship Foundation for its support of his doctoral research.

References

[1] Goldberg, D. E., “Genetic Algorithms in Search”, Optimization & Machine Learning, Boston, MA: Addison-Wesley (1989).

[2] Holland, J., “Adaptation in Natural and Artificial Systems”, Ann Arbor: The University of Michigan Press (1975).

[3] Michalewicz, Z., “Genetic Algorithms + Data Structures = Evolution Programs”, Springer-Verlag (1996).

[4] Schwefel, H. P., “Evolutions Strategie und Numerische Optimierung”, Ph.D. thesis (1975).

[5] Schwefel, H. P., “Numerical Optimization of Computer Models”, Wiley Chichester (1981).

[6] Schwefel, H. P., “Evolution Optimum Seeking”, Sixth Generation Computer Technology Series, John Wiley and Sons (1995).

[7] Fogel, L. J., “Autonomous Automata”, Industrial Research, Vol.4, pp.14-19 (1962).

[8] Koza, J., “Genetic Programming”, On the Programming of Computers by Means of Natural Selection, The MIT Press (1992).

[9] Fogel, D. B., “System Identification Trough Simulated Evolution”, A Machine Learning Approach, USA: Ginn Press (1991).

[10] Wong, K.P., Li A., and Law T.M.Y., “Advanced constrained genetic algorithm load flow method”, Generation, Transmission and Distribution, IEE Proc., Vol.146, No.6, pp.609-616 (1999).

[11] Buchtala, O., Klimek, M., and Sick, B., “Evolutionary Optimization of Radial Basis Function Classifiers for Data Mining Applications”, IEEE Trans. on Syst. Man, Cybern., Part B, Vol.35, No.5, pp.928-947 (2005).

[12] Sheu S.-T. and Chuang Y.-R., “A Pipeline-Based Genetic Algorithm Accelerator for Time-Critical Processes in Real-Time Systems”, IEEE Trans. on Computers, Vol.55, No.11, pp.1435-1448 (2006).

[13] Wong K.P., Li A., and Law T.M.Y., “Development of Constrained-Genetic-Algorithm Load-Flow Method”, IEE Proc., Generation, Transmission and Distribution, Vol.144, pp.91-99 (1997).

[14] Song Y.H., and Chou C.S.V. , “Advanced Engineered-Conditioning Genetic Approach to Power Economic Dispatch”, IEE Proc., Gener. Transm. Distrib., Vol.144, No.3, pp.285-292 (1997).

[15] Potter W.D., Miller J.A., Tonn B.E., Gandham R.V., and Lapena C.N., “Improving the reliability of heuristic multiple fault diagnosis via the EC-based genetic algorithm”, J. of Applied Intelligence, Vol.2, pp.5-23, Springer (1992).

[16] Miller J.A., Potter W.D., Gandham R.V. and Lapena C.N., “An Evaluation of Local Improvement Operators for Genetic Algorithms”, IEEE Trans. on Syst. Man, Cybern., Vol.23 pp.1340-1351 (1993).

[17] David Edward Goldberg , “Genetic Algorithms in Search, Optimization, and Machine Learning”, Reading MA: Addison-Wesley, (1989).

[18] Kim J.-H., Chae H.-K., Jeon J.-Y., and Lee S.-W., “Identification and Control of Systems with Friction using Accelerated Evolutionary Programming”, IEEE Control Systems Magazine, Vol.16, pp 38-47 (1996).

[19] Bhandaarl D., and Murthy C. A., “Genetic Algorithm with Elitist Model and Its Convergence”, Int. J. Pattern Recognition and Artificial Intelligence, Vol.10, No.5, pp.731-734 (1996).

[20] Zitzler E., Deb K., and Thiele L., “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results” Evolutionary Computation, Vol.8, No.2, pp.173-195 (2000).

[21] Cui S., Mohan A., and Weile D.S., “Pareto Optimal Design of Absorbers Using a Parallel Elitist Nondominated Sorting Genetic Algorithm and the Finite Element-Boundary Integral Method”, IEEE Trans. on Antennas and Propagation, Vol.53, No.6, pp.2099-2107 (2005).

[22] Carretero J.A., and Nahon M.A., “Solving Minimum Distance Problems with Convex or Concave Bodies Using Combinatorial Global Optimization Algorithms”, IEEE Trans. on Syst. Man, Cybern., Part B, Vol.35, No.6, pp.1144 - 1155 (2005).

[23] Mantawy A.H., Abdel-Magid Y.L. and Selim S.Z., “Integrating Genetic Algorithms, Tabu Search, and Simulated Annealing for the Unit Commitment Problem”, IEEE Trans. on Power Systems, Vol.14, No.3, pp.829-836 (1999).

[24] Mehrdad Hakimi-Asiabara, Seyyed Hassan Ghodsypoura, and Reza Kerachianb, “Multi-Objective Genetic Local Search Algorithm Using Kohonen’s Neural Map”, Computers and Industrial Engineering, Vol.56, No.4, pp.1566-1576 (2009).

[25] Lahiri A., and Chakravorti S., “Electrode-Spacer Contour Optimization by ANN Aided Genetic Algorithm”, IEEE Trans. on Dielectrics and Electrical Insulation, Vol.11, No.6, pp.964-975 (2004).

[26] Hajji O., Brisset S., and Brochet P., “A Stop criterion to Accelerate Magnetic Optimization Process Using Genetic Algorithms and Finite Element Analysis”, IEEE Trans. on Magnetics, Vol.39, No.3, pp.1297-1300 (2003).

[27] Beasley D., Bull D. R., and Martin R. R., “An Overview of Genetic Algorithms, Part 1: Fundamentals”, Univ. Comput., Vol.15, pp.58-69 (1993).

[28] Rudolph G., “Convergence Analysis of Canonical Genetic Algorithms”, IEEE Trans. on Neural Networks, Vol.5, pp.96-101 (1994).

[29] Grefenstette J. , “Optimization of Control Parameters for Genetic Algorithms”, IEEE Trans. on Syst. Man, Cybern., Vol.16, pp.122-128 (1986).

[30] Zhou Z. Z., Ong Y. S., Nair P.B., Keane J.A., and Lum Y. K. , “Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization”, IEEE Trans. on Syst. Man, Cybern., Part C: Applications and Reviews, Vol.37, No.1, pp.66-76 (2007).

[31] Wang Y., Cai Z.X., Guo G.Q., and Zhou Y.R., “Multiobjective Optimization and Hybrid Evolutionary Algorithm to Solve Constrained Optimization Problems”, IEEE Trans. on Syst. Man, Cybern., Part B, Vol.37, No.3, pp.560-575 (2007).

[32] Takagi H., Ingu T., and Ohnishi K., “Accelerating a GA Convergence by Fitting a Single-Peak Function”, J. of Japan Society for Fuzzy Theory and Intelligent Informatics, vol.15, no.2, pp.219-229 (2003).

[33] Goldberg D. E., Korb B., and Deb K., “Messy Genetic Algorithms: Motivation, Analysis, and First Result”, Complex System, Vol.3, 493-530 (1988).

[34] Whitley D., Mathias K., and Fitzhorn P., “Delta Coding: an Interative Search Strategy for Genetic Algorithms”, Proc. Of 4th Int. Conf. on Genetic Algorithms, pp.77-84, San Mateo, CA (1991).

[35] Schraudolph N. N., and Belew R. K., “Dynamic Parameter Encoding for Genetic Algorithms”, Machine Learning, Vol.9, No.1, pp.9-22 (1992).

[36] Eshelman L., and Shaffer J. D., “Real-coded Genetic Algorithms and Interval Schemata”, In Whitley(Ed.), Foundations of Genetic Algorithm 2, Los Altos, CA, pp.187-202, Morgan Kaufmann (1993).

[37] Zadeh Lotfi A., “Fuzzy Logic, Neural Networks, and Soft Computing”, Communications of the ACM, Vol.37 No.3, pp.77-84 (1994).

[38] Hideyuki Takagi, “Introduction to fuzzy systems, neural networks, and genetic algorithms”, in Intelligent Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms, Ch.1, pp.1-33, edited by D. Ruan, Kluwer Academic Publishers (Norwell, Massachusetts, USA), (September, 1997).

[39] Hideyuki Takagi, “Fusion Technology of Neural Networks and Fuzzy Systems: A Chronicled Progression from the Laboratory to Our Daily Lives”, International Journal of Applied Mathematics and Computer Science, Vol.10, No.4, pp.647-673 (2000).

[40] Pei Y. and Takagi, H., “Accelerating Evolutionary Computation with Elite Obtained in Projected One-Dimensional Spaces”, 5th Int. Conf. on Genetic and Evolutionary Computing, Kimmen Taiwan (ICGEC2011), accepted (Aug. 29-Spt. 1, 2011).