A Hybridization of Adaptive Genetic Algorithm and Particle Swarm Optimization for Numerical Optimization Functions

YoungSu Yun* and Mitsuo Gen**

* Division of Business Administration, Chosun University,
375 Seosuk-dong, Dong-gu, Gwangju 501-759, Republic of Korea
Tel:+82 62 230 6243, Fax: +82 62 226 9664 E-mail: ysyun@chosun.ac.kr
** Graduate School of Information, Production & Systems, Waseda University,
Kitakyushu, 808-0135, Japan

Abstract

Heuristic optimization using hybrid algorithms have provided a robust and efficient approach for solving many optimization problems. In this paper, a new hybrid algorithm using adaptive genetic algorithm (aGA) and particle swarm optimization (PSO) is proposed. The proposed hybrid algorithm is applied to solve numerical optimization functions. The results are compared with those of GA and other conventional PSOs. Finally, the proposed hybrid algorithm outperforms others.

Keywords: Heuristic optimization, hybrid algorithm, genetic algorithm, particle swarm optimization.

1. Introduction

In the last two decades, there has been a much more interest in the fields of evolutionary algorithms. One of evolutionary algorithms, genetic algorithm (GA) is the best-known branch. GA has a stochastic search procedure based on the mechanism of natural selection, evolution and genetics [Goldberg, 1998].

GA, differing from conventional stochastic search algorithms, start with an initial set of random solutions called population. Each individual in the population represent a solution to the problem under consideration. The individuals evolve through successive iterations called generations. During each generation, the individuals are evaluated using some measures of fitness and are changed by genetic operator such as crossover, mutation and selection. The search process is continued until a pre-determined number of generations is reached or the global optimal solution of the problem is located (Gen and Cheng 1997).

By the unique search mechanism of GA, it has proved to be a versatile and effective approach for solving many optimization problems. Nevertheless, there are several situations where simple GA does not well perform particularly. First, GA has a weakness in taking too much time to adjust fine-tuning structure of GA parameters, i.e., crossover rate, mutation rate, and others. Secondly, once the optimal solution region is identified by GA search process, finding the global optimal solution using simple GA becomes inefficient or impossible due to the random nature of GA search. This kind of "blindness" may prevent it from being really of practical interest for lot of applications.

To overcome the first weakness mentioned above, various adaptive techniques to adjust the fine-tuning structure of the parameters used in GAs have been developed (Srinivas and Patnaik, 1994; Wu *et al.*, 1998; Mak *et al.*, 2000; Yun, 2002). Srinivas and Patnaik (1994) controlled the rates according to the fitness values of the population at each generation. Mak, Wong, and Wang (2000) adaptively controlled the crossover and mutation rates according to the performance of GA operators in a manufacturing cell formulation problem.

The genetic parameters controlled by adaptive techniques adaptively regulated during genetic search process. Therefore, much time for the fine-tuning of the parameters can be saved, and the GA search ability can be improved in finding a global optimum.

For the second weakness, various methodologies for hybridization of GA and other optimization algorithms have been developed. Davis (1991) and Ishibuchi (1994) suggested random generate and test algorithm and multi-start descent algorithm for initializing GA populations respectively. Li and Jiang (2000) presented a new stochastic approach SAGACIA based on proper integration of simulated annealing (SA) algorithm, GA, and chemotaxis algorithm (CIA) for solving complex optimization problems.

Recently, as a new type of hybridization, particle swarm optimization (PSO) has been incorporated into GA. PSO proposed by Kennedy and Eberhart (1995) follows from observations of social behaviors of animal, such as bird flocking and fish schooling. The theory of PSO describes a solution process in which each particle flies through the multidimensional search space while the particle's velocity and position are constantly updated according to the best previous performance of the particle or of the particle's neighbors, as well as the best performance of the particles in the entire population (Kao and Zahara 2008).

Compared with GA, PSO has some attractive characteristics. It has memory, so knowledge of good solutions is retained by all the particles; on the other hand in GA, previous knowledge of the problem is discarded once the population changes. It has constructive cooperation between particles; that is, particles in the swarm share information among themselves. Therefore, PSO has been successfully applied to various numerical optimization functions (Kennedy et al., 2001).

By the compensatory property between GA and PSO mentioned above, we will propose a new hybrid algorithm using adaptive GA (aGA) and PSO denoted as aGA-PSO. The performance of the proposed aGA-PSO will be compared with those of GA and conventional GA-PSO using several numerical optimization functions.

2. Adaptive Genetic Algorithm and Particle Swarm Optimization

2.1. Adaptive Genetic Algorithm (aGA)

GA has been known to offer significant advantages over conventional methods by using simultaneously several search principles and heuristics (Goldberg, 1998). The most important ones include a population-wide search, a continuous balance between convergence and diversity, and the principle of building-block combination. Despite of the strongpoint in GA application, it has a weakness in taking too much time to adjust fine-tuning structure of GA parameters. Therefore, aGAs with various types of adaptive schemes have been developed (Srinivas and Patnaik, 1994; Wu *et al.*, 1998; Mak *et al.*, 2000; Yun, 2002).

The basic logic of adaptive scheme is to enhance the performance of the search by adaptively regulating GA parameters during genetic search process. Whenever a new offspring is added to the population, a pointer is established for the genetic operator that generates the offspring. A check is then made to determine if the fitness of the offspring is better or worse than its parents. The percentage of improvement or degradation is recorded, and this record is reserved for later adjustments of the occurrence rates of GA operators.

2.2 Particle Swarm Optimization (PSO)

PSO, one of the latest evolutionary optimization techniques, is developed by Kennedy and Eberhart (1995). PSO concept is based on a metaphor of social interaction such as bird flocking and fish schooling. The particles, which are potential solutions in PSO algorithm, fly around in the multi-dimensional search space and the positions of individual particles are adjusted according to its previous best position, and the neighborhood best or the global best solutions. (Kao and Zahara, 2008)

Since all particles in PSO are kept as members of the population during the searching process, PSO is the only evolutionary algorithm that does not implement survival of the fittest. As simple and economic in concept and computational cost, PSO has been known to successfully optimize a wide range of continuous optimization problems (Yoshida et al. 2000; Brandstatter and Baumgartner, 2002).

3. Hybrid algorithm using aGA and PSO

Several hybrid algorithms using GA and PSO have been proposed in literatures (Settles and Soule 2005; Holden and Freitas 2007).

Recently a new concept of hybrid algorithm using GA and PSO, denoted as GA-PSO, was proposed by Kao and Zahara (2008). In their paper, they used 4N individuals. Among them, the 2N individuals with best fitness values are adopted to GA search process and the remaining 2N individuals with worst fitness values are used to PSO search process. During the search process of the GA-PSO, GA used its operators, i.e., the crossover using 100% rate and the mutation using 20% rate. Especially, two crossover and mutation operators were developed. For PSO, conventional updating scheme was used and it regulates the velocity and position of particles. This procedure is stopped until a termination criterion is satisfied. The unique characteristics of the GA-PSO are the use of 4N individuals and the developed new crossover and mutation operators.

Therefore, as a similar concept we will propose a new hybrid algorithm using aGA and PSO called aGA-PSO in this paper. For the aGA, adaptive crossover and mutation operators are used. That is, the occurrence rates of the two operators are adaptively regulated by adaptive scheme during the genetic search process. For this logic, we use the concept of Mak, Wong, and Wang (2000). They employed the fitness values of parent and offspring at each generation in order to construct adaptive crossover and mutation operators. The procedure of the adapting strategy is as shown in Figure 1.

```
Procedure: Adaptive scheme for crossover and mutation
           operators
  begin
    if (f̄_par_size(t) / f̄_off_size(t)) ≥ 0.1 then
         p_C(t+1) = p_C(t) + 0.01,  p_M(t+1) = p_M(t) + 0.005;
    if (f̄_par_size(t) / f̄_off_size(t)) ≤ 0.1 then
         p_C(t+1) = p_C(t) - 0.01,  p_M(t+1) = p_M(t) - 0.005;
    if -0.1 < (f̄_par_size(t) / f̄_off_size(t)) - 1 < 0.1 then
         p_C(t+1) = p_C(t),  p_M(t+1) = p_M(t);
    end
end
```

Figure 1. Procedure of Adaptive scheme

In Figure 1, *par_size* and *off_size* are the parent size and offspring size satisfying constraints, respectively. $\overline{f_{par\_size}}(t)$ and $\overline{f_{off\_size}}(t)$ are respectively the average fitness values of parents and offspring at generation t. $p_C(t)$ and $p_M(t)$ are the rates of crossover and mutation operators at generation t, respectively. In the cases of $(\overline{f_{par\_size}}(t)/\overline{f_{off\_size}}(t)) \geq 10\%$ and $(\overline{f_{par\_size}}(t)/\overline{f_{off\_size}}(t)) \leq 10\%$, the adjusted rates should not exceed the range from 0.0 to 1.0 for $p_C(t+1)$ and $p_M(t+1)$.

Table 1. Four types of numerical optimization functions

Fun. Name	Description	Global Optimum
F1	*minimize* $F=(x_1-x_2)^2+((x_1+x_2-10)/3)^2$	$(x_1,x_2)=(5,5), F=0$
F2	*minimize* $F=100(x_1^2-x_2)^2+(1-x_1)^2$	$(x_1,x_2)=(1,1), F=0$
F3	*minimize* $F=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$	$(x_1,x_2)=(0,0), F=0$
F4	*minimize* $F=-\cos(x_1)\cos(x_2)\exp(-((x_1-\pi)^2+(x_2-\pi)^2))$	$(x_1,x_2)=(\pi,\pi), F=-1$

The above-stated procedure is evaluated in all generations during genetic search process, and the occurrence rates of crossover and random mutation operators are adaptively regulated according to the result of the procedure.

For the PSO, the basic concept of suggested in Kao and Zahara (2008) is used.

Using the aGA and PSO concepts described above, the aGA-PSO is developed. Its detailed implementation procedure is as follows:

Step 1. Initialization: Generate a population of 4N size for an N-dimensional problem.

Step 2. Fitness evaluation and ranking: Evaluate the fitness of each of the 4N individuals. Rank them on the basis of the fitness values.

Step 3. aGA method: apply aGA procedure using the 2N best individuals.

Step 3-1. (Selection): From the population, select the 2N best individuals.

Step 3-2. (Adaptive crossover and mutation): For crossover and mutation operators, uniform arithmetic crossover operator (Michalewicz, 1994) and uniform mutation operator (Michalewicz, 1994) are used, respectively.

The crossover and mutation rates are regulated by the procedure of the adaptive scheme shown in Figure 1.

Step 4. PSO method: apply PSO operators (velocity and position of particles) using the 2N worst individuals.

Step 4-1. (Velocity of particle): the updating scheme of the velocity of each particle is as following.

$$v^i_{k+1}=w\cdot v^i_k+C_1D_1(lbest_k-x^i_k)+C_2D_2(gbest-x^i_k) \quad (1)$$

where w = 0.5 + (*rand*[0,1]/2.0), C_1=C_2=2.0, D_1=D_2=*rand*[0,1]. $lbest_k$ is the best fitness value at k-th iteration. *gbest* is the best fitness value at all iteration. v_k is the velocity of the i-th particle at iteration k. x^i_k is the position of the i-th particle at iteration k.

Step 4-2. (Position of particle): the updating scheme of the position of each particle is as following.

$$x^i_{k+1}=x^i_k+v_{k+1}i \quad (2)$$

Step 5. Termination condition: if either a predefined-termination condition is satisfied or global optimal solution already known is located, then all steps are stooped. Otherwise, go Step 2.

Figure 2 shows the flow chart of the proposed aGA-PSO.

4. Numerical Examples

In this section, four types of numerical optimization functions with their global optimal solutions already known are presented in order to prove the effectiveness of the proposed aGA-PSO. They are summarized in Table 1.

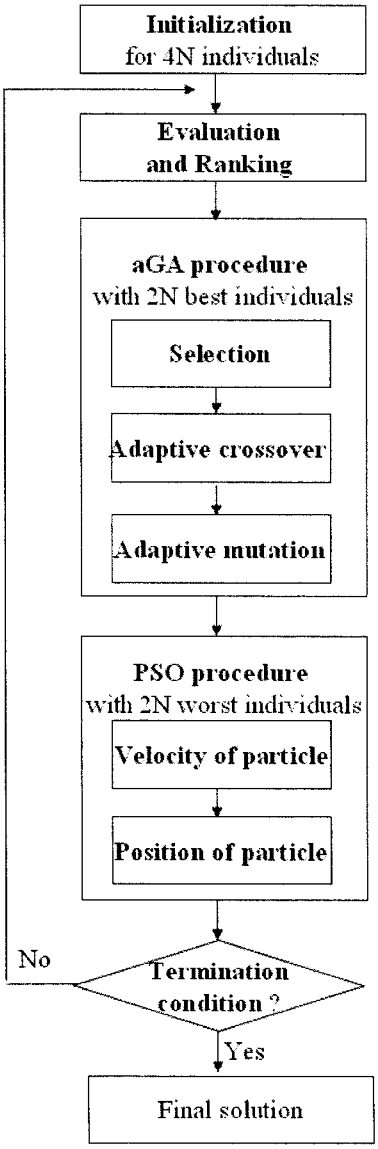

Figure 2. Flow chart of aGA-PSO.

For various comparisons, several conventional algorithms are presented: i) a simple GA (sGA), ii) the aGA using the scheme of Mak, Wong, and Wang (2000) and iii) the GA-PSO developed by Kao and Zahara (2008).

Each algorithm (sGA, aGA, GA-PSO and the proposed aGA-PSO) is compared with each other using various measures of performance as shown in Table 2.

Table 2. Measures of performance

Notation	Description
ANG	Average number of generations until termination condition is satisfied.
NGS	Total number of getting stuck at a local optimal solution
CPU	Average CPU time (unit: Sec.)

For experimental comparison under a same condition, the parameters used in each algorithm are set as follows: maximum iteration number is 2,000, population size 20, crossover rate 0.5, mutation rate 0.05. Altogether 20 iterations are executed to eliminate the randomness of the searches in each algorithm. The procedures of each algorithm are implemented in Visual Basic language under IBM-PC Pentium IV computer with 1.2Ghz CPU speed and 1GB RAM. The computational results of each algorithm using the four types of numerical optimization functions in Table 1 are shown in Tables 3, 4, 5, and 6.

Table 3. Computation result for F1

	ANG	NGS	CPU
sGA	1,670	15	5.50
aGA	789	6	5.12
GA-PSO	35	0	8.54
aGA-PSO	20	0	8.90

Table 4. Computation result for F2

	ANG	NGS	CPU
sGA	1,200	12	9.12
aGA	916	4	8.09
GA-PSO	350	0	14.33
aGA-PSO	265	0	13.01

Table 5. Computation result for F3

	ANG	NGS	CPU
sGA	2,000	20	11.34
aGA	1,800	10	10.44
GA-PSO	980	2	20.21
aGA-PSO	705	0	18.87

Table 6. Computation result for F4

	ANG	NGS	CPU
sGA	2,000	20	13.50
aGA	1,440	8	12.43
GA-PSO	1,200	5	23.54
aGA-PSO	890	0	21.90

In Tables 3, 4, 5 and 6, the computation result of sGA are the worst in terms of ANG and NGS. Especially, in Tables 5 and 6, it always gets stuck at local optimal solutions, which means that sGA itself may not be difficult to find global optimal solution. Therefore, any adaptive scheme or a hybrid concept using other optimization algorithms is required to improve the performance of sGA.

The computation results using aGA show any merit against sGA in terms of the ANG, NGS and CPU, which implies that the adaptive scheme used in aGA can improve the performance of sGA. However, when compared with GA-PSO and aGA-PSO, aGA requires additional hybrid concept though it has adaptive scheme.

With adaptive scheme and a hybrid concept, GA-PSO and aGA-PSO have better performance in terms of ANG and NGS than sGA and aGA, since the formers have PSO as an additional scheme in GA. However, in terms of the CPU, the search speeds of GA-PSO and aGA-PSO are slower than those of sGA and aGA because the formers have 4N individuals, but the letters 2N individuals.

Especially, in comparison between GA-PSO and aGA-PSO, the latter outperforms the former in terms of ANG, NGS and CPU of Tables 3, 4, 5 and 6. This implies that the adaptive scheme used in aGA-PSO is efficient in locating the global optimal solutions and can reduce the search speed during its search process.

5. Conclusion

In this paper, aGA-PSO has been proposed as a new approach for locating the global optimal solutions of numerical optimization functions. The proposed aGA-PSO has an adaptive scheme to adaptively regulate crossover and mutation operators and cooperates with PSO for making hybrid algorithm.

Four types of numerical optimization functions have been used to prove the efficiency of aGA-PSO. The performance of aGA-PSO has been compared with those of sGA, aGA and conventional GA-PSO using some measures of performance. Various comparison results have shown that most of the performances of aGA-PSO are superior to the competing algorithms (sGA, aGA, GA-PSO).

Reference

Brandstatter, B. and Baumgartner, U. (2002) Particle Swarm Optimization Mass-spring System Analog, *IEEE Transactions on Management*. Vol. 38, pp.997–1000.

Davis L. (1991) *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

Gen, M. and Cheng, R. (1997) *Genetic Algorithms and Engineering Design*, John-Wiley & Sons.

Goldberg, D.E. (1998) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, NY.

Holland, J. (1997) *Adaptation in Natural and Artificial Systems*, University of Michigan Press.

Holden, N. P. and Freitas, A. A. (2007) A hybrid PSO/ACO Algorithm for Classification, *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation (GECCO 2007).*

Ishibuchi, H., Yamamoto, N., Murata, T. and Tanaka, H. (1994) Genetic Algorithm and Neighborhood Search Algorithms for Fuzzy Flowshop Scheduling Problems, *Fuzzy Sets and Systems*, Vol. 67, pp.81-100.

Kao, T-T and Zahara, E. (2008) *A Hybrid Genetic Algorithm and Particle Swarm Optimization for Multimodal Functions*, Applied Soft Computing, Vol. 8, pp. 849-857.

Kennedy, J. Eberhart, R.C. and Shi, Y. (2001) *Swarm Intelligence*, Morgan Kaufmann Publisher, San Francisco.

Kennedy, J. and Eberhart, R. C. (1995) Particle Swarm Optimization, *Proceedings on IEEE International Conference on Neural Networks*, Piscataway, NJ, USA, pp. 1942–1948.

Li, B. and Jiang, W. (2000) A Novel Stochastic Optimization algorithm, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 30, No.1, pp. 193-198.

Mak, K. L., Wong, Y. S. And Wang, X. X. (2000). An Adaptive Genetic Algorithm for Manufacturing Cell Formation, *International Journal of Manufacturing Technology*, Vol.16, pp.491-497.

Michalewicz, Z. (1994) *Genetic Algorithms + Data Structures = Evolution Program*, Second Extended Edition, Spring-Verlag.

Settles, M. and Soule, T. (2005) Breeding Swarms: a GA/PSO Hybrid, *Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO 200.)*

Srinvas, M. and Patnaik, L. M. (1994). Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms, *IEEE Transaction on Systems, Man and Cybernetics*, Vol. 24, No. 4, pp.656-667.

Wu, Q. H., Cao, Y. J. and Wen, J. Y. (1998) Optimal Reactive Power Dispatch using an Adaptive Genetic Algorithm, *Electrical Power and Energy Systems*, Vol. 20, No. 8, pp. 563-569.

Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S. and Nakanishi, Y. (2000) A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment, *IEEE Transactions on Power Systems*, Vol. 15, pp. 1232–1239.

Yun, Y. S. (2002) Genetic Algorithm with Fuzzy Logic Controller for Preemptive and Non-preemptive Job Shop Scheduling Problems, *Computers and Industrial Engineering*, Vol. 43, No. 3, pp. 623-644.