Available online at www.elixirpublishers.com (Elixir International Journal)

Applied Mathematics

Elixir Appl. Math. 39 (2011) 4814-4817

ISSN: 2229-712X

A hybrid genetic algorithm with effective local search technique

Jinghui Gao[a], Rui Shan[a], Hongfang Cui[a,b] and Lingling Li[a,b]

College of Science, Yanshan University, Qinhuangdao066004, Hebei in China P.R.

ARTICLE INFO

Article history:

Received: 13 July 2011;

Received in revised form:

21 September 2011;

Accepted: 28 September 2011;

Keywords

Hybrid genetic algorithm,

Local search technique,

The steepest descent method.

Classic Number: TP301; O224.

ABSTRACT

This paper proposes a hybrid genetic algorithm (HGA) with local search technique. For designing the HGA, a local search technique is incorporated in the loop of genetic algorithm (GA), and whether or not the local search technique is used in the GA is automatically determined by the local search scheme. A local search scheme is developed in this paper. which is to use the conditional local search method that can measure the average fitness values obtained from the continuous two generations of the HGA. To prove the efficiency of the algorithm, a basic GA (BGA) with local search technique and a hybrid GA (HGA) with local search technique and are also presented. In numerical example, both the algorithms (BGA, HGA) are tested and analyzed. Finally, the efficiency of the proposed HGA is proved by various measures of performance.

Introduction

Genetic algorithm (GA) has been successfully used in providing optimal or near optimal solutions to many optimization problems. Especially, GA has been also used for solving sophisticated optimization problems with complex search spaces and many constraints. With the development of various modern intelligent algorithms, an increasing tendency in merging GA and fuzzy logic or other heuristic methods is upward. As a result, much more attentions have been focused on the design of various hybrid genetic algorithms (HGAs) in [1] and in [2].

The generalized components of the HGA developed so far are to combine GA with various local search techniques such as the steepest descent method and exhaustive search ones in [3]. It has been known that most of the HGAs have better performance than GA, since the formers can overcome the weaknesses of GA such as the premature convergence of solution and the absence of local search technique. Therefore, recent HGAs have been proved to be more efficient than GA in the applications to many optimization problems such as engineering design problems, reliability optimization problems and network design problems etc. In these applications to many optimization problems, however, the most of the local search techniques used were problem specifics and were designed using trial-and-error experimentation without any generalization or analysis with respect to their convergence characteristics and reliabilities. For improving these weaknesses in the application of local search techniques, a local search technique can be an alternative, since it can automatically control whether the local search technique is used in GA loop or not. Therefore, any generalization or analysis in applying the local search technique to GA loop is not required.

There has been a little study on the local search technique with some technique in GA. It suggested a concept of the local search technique with adaptive technique in GA loop in [4]. Their concept is to apply a local search technique to GA loop only when GA performance is changed by the relative coefficient of variation of the fitness functions between generations in GA.

The local search technique used is the steepest descent method suggested in [8], and it is automatically controlled by the two adaptive local search schemes mentioned above. For various comparisons with the proposed HGA, two competing algorithms such as a basic GA (BGA) and the conventional HGA with local search technique are also presented.

In Section 2, a methodology for constructing the HGAs with GA and local search technique are suggested, and then the detailed descriptions and logics of local search technique is proposed. The implementation procedure of the proposed HGA using local search technique is also appeared. Two numerical examples are presented to prove the efficiency of the proposed HGA in Section 2. Finally a conclusion is followed.

Design of HGAs with Effective Local Search Schemes

In this section, the methodology for constructing the HGA is mentioned. First the basic concepts and implementation procedures of GA and local search technique are suggested. Second, as a main part of this study, the concepts and detailed implementation procedures of the local search technique are proposed.

Genetic Algorithm

The main role of GA approach is to perform global search within all feasible search spaces. For this purpose, GA has particular mechanisms: (i) population-oriented search technique for locating more various individuals, (ii) three genetic operators (selection, crossover and mutation) for more various changes within GA population.

In the representation of GA, we use real-number representation instead of bit-string one. Real-number representation has several advantages of being better adapted to numerical optimization problems with continuous design variables and of speeding up the search over the bit-string representation, and of easing the development of approaches for hybridizing with other conventional methods such as local search technique.

Tele:

E-mail addresses: gjhlove123@163.com

The proposed GA is used as the main algorithm of the HGAs and the detailed heuristic procedure for implementing GA is as follows:

Step 1: Initial population: Generate initial population randomly.

Step 2: Genetic operators.

Selection: elitist strategy in enlarged sampling space in [5].

Crossover: uniform arithmetic crossover operator in [5].

Mutation: uniform mutation operator in [6].

Step 3: Evaluation: Do fitness test using the offspring satisfying constraints.

Step 4: Stop condition: If a pre-defined maximum generation number is reached or an optimal solution is located during genetic search process, then stop; otherwise, go to Step 2.

The steepest descent method

GA can do global search in entire space, but there is no way for exploring the search space within the convergence area generated by GA loop. Therefore, it is sometimes impossible or insufficient for GA to locate an optimal solution in the optimization problems with complex search spaces and constraints. To overcome this weakness, various methods for hybridizing GA using conventional local search techniques have been suggested in [2]. One of the common forms of hybrid GA is to incorporate a local search technique to GA loop. With this hybrid approach, local search technique is applied to each newly generated offspring to move it to a local optimal solution before injecting it into the new population.

In our approach, as local search technique, the steepest descent method is incorporated in GA loop. Therefore, GA carries out global search and the steepest descent method carries out local search around the convergence area by GA loop. The latter can guarantee the desired properties of local search technique for hybridization with GA. Its detailed procedure, when minimization is assumed, is given as follows:

Procedure: The steepest descent method in GA loop

Select an optimal individual x as initial iteration point in current GA loop;

Repeat

Linearly search in the local neighborhood of x;

$$x_{k+1} = linear - search(x_k, \cdots - f(x_k))$$

Calculate

$$f(x_{k+1}), \cdots, f(x_{k+1})$$

Select the individual x_{k+1} among the new individuals;

Until (satisfy convergence conditions);

Compare all the optimal solutions;

End

Local search technique

The basic concept of applying local search techniques to GA is to consider whether GA is converging to global optimal solution or not. These two situations can be summarized in Figs. 1 and 2. When GA is converging to global optimal solution like Fig.1, its solution is continuously improved. However, when GA is not converging to global optimal solution like Fig. 2, the performance of GA definitely deteriorates. If this situation continuously proceeds, it may be difficult for GA search to avoid premature convergence to a local optimal solution. The technique that helps improving this situation is to insert new individuals with certain high fitness values into current GA loop. A local search technique that can search around the convergence area by GA loop is a possible alternative, since it can generate new individuals having certain high fitness values like the superior individuals generated by GA.

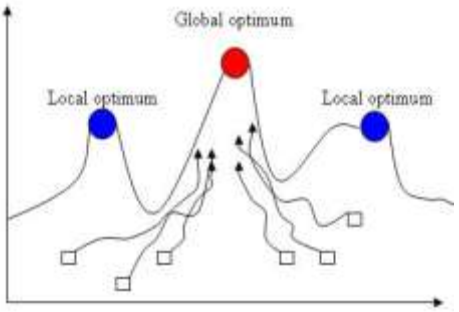

Fig.1. Situation that GA is converging

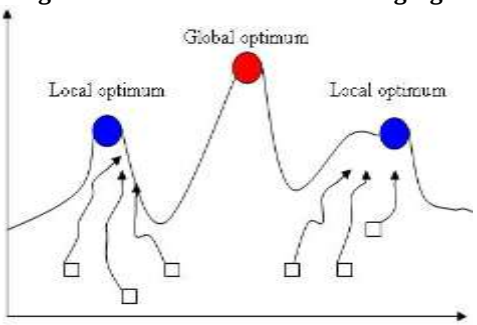

Fig.2. Situation that GA is not converging

Based on the concept mentioned above, a local search techniques are proposed, which can automatically control whether the steepest descent method suggested in Section 2.3.1 is used in GA loop or not.

Local search technique details

The local search technique is to apply the steepest descent method that is conditionally used in GA loop. It is adapted in response to the recent performances obtained from genetic search process to optimal solution. For this technique, the average fitness values resulting from continuous two generations of GA loop are used, and then the fitness value ratio (FVR) for the next generation is calculated as follow:

$$FVR(t) = \frac{\overline{f_{new-pop(t)}}}{\overline{f_{new-pop(t-1)}}} \qquad (1)$$

where $f_{new-pop(t)}$: average fitness value of the new population resulting from elitist selection strategy using parent and offspring populations at generation t. By the value of the $FVR(t)$, whether the steepest descent method is used or not will be automatically determined in each GA loop. The technique, when minimization is assumed, is as follows:

$$\begin{cases} apply\ steepest\ descent\ method\ to\ GA\ loop, \\ apply\ GA\ only, \end{cases} \quad \begin{array}{l} if\ FVR(t) < 1 \\ otherwise \end{array} \qquad (2)$$

The first condition of the Eq. (2) explains that the steepest descent method is applied to GA loop, if the average fitness value of a current generation is higher than that of the previous one, which means that since the current situation at the generations t and $t-1$ is not converging to optimal solution at all, we should insert a local search technique into GA loop so that the convergence ability to optimal solution can be reinforced. On the other hand, in the second condition of the Eq.

(2), we use GA only since the current situation at the generations t and $t-1$ is well converging to optimal solution.

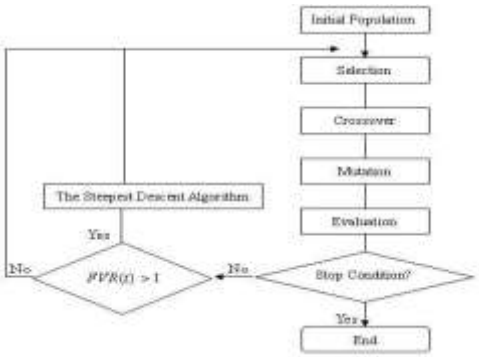

Fig.3 Flow chart of HGA

Numerical examples

In this section, two test problems are used to compare the performances of the BGA and HGA. For more various comparisons, basic GA (BGA) with local search technique and the hybrid GA (HGA) with local search scheme are also performed. Both algorithm are compared with each other using various measures of performance as shown in Table 2.

In Table 2, the BFV, AFV, ANG and CPU are respectively obtained after each algorithm reaches to a predefined iteration number (in our case, 20 iterations). The NGS means the total number that each algorithm gets stuck at a local optimal solution, instead of finding the global optimal solution.

For experimental comparison under a same condition, the parameters used in each algorithm are set as follows: maximum generation number is 5000, population size 20, crossover rate 0.5, and mutation rate 0.1, search range for the steepest descent method 0.6. 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 III computer with 800 MHz CPU speed and 1GB RAM.

Test problem 1

$$f(x_1,x_2)=0.5-\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{(1+0.001(x_1^2+x_2^2))^2},\quad -100\le x_1,x_2\le 100$$

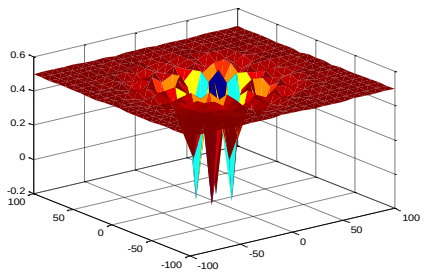

Fig.4 The camber of f(x)

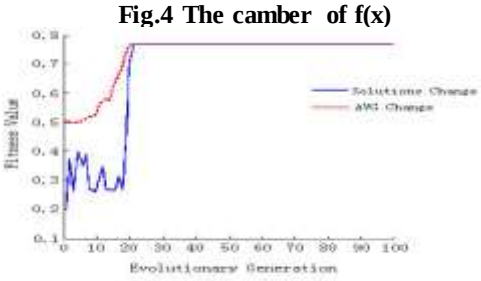

Fig.5 Curve chart of solutions change using BGA with local search technique

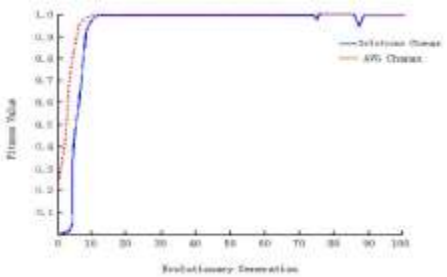

Fig.6 Curve chart of solutions change using HGA.

The computational results have shown that the proposed HGA with local search technique is more efficient in most of the performance comparisons than the BGA. Especially, the latter has good convergence.

Test problem 2

The problem is a simple optimal redundancy allocation one with 15 subsystems, the detailed mathematical formulation is as follows:

$$\max . f(x)=\prod_{j=1}^{15}\{1-(1-r_j)^{x_j}\}$$

$$s.t. g_1(x)=\sum_{j=1}^{15}(c_j\cdot x_j)\le 400$$

$$g_2(x)=\sum_{j=1}^{15}(w_j\cdot x_j)\le 414$$

$$1\le x_j\le 10:\text{int}\,eger\quad j=1,...,15$$

where, $f(x)$: system reliability, x_j: decision variable that is the number of units in the j th subsystem, $g_1(x)$: total cost for the system, $g_2(x)$: total physical limi -tation for the system, w_j: physical limitation (size or weight, etc.) of the j th subsystem, r_j and c_j: reliability and cost of a unit to be used the in jth subsystem, respectively.

The pre-dened constant coefficients for the problem are shown in Table1. The optimal solution was already known as the optimal value $f(x)=0.9456$ with

$$x=[3\quad 4\quad 5\quad 3\quad 3\quad 2\quad 4\quad 5\quad 4\quad 3\quad 3\quad 4\quad 5\quad 5\quad 5]$$

Conclusion

In this paper, the hybrid GA with local search technique has been proposed. For the proposed GA, local search technique for automatically controlling the use of the local search technique in GA loop have been designed. The local search technique measures the average fitness values resulting from the continuous two generations in the GA loop.

The steepest descent method has been used for the local search technique. For more various comparisons, we have also presented the canonical GA (BGA) and the HGA with local search technique.

Both the algorithms (BGA, HGA) have been tested and analyzed using two test problems in numerical examples. Various measures of performance such as best fitness value, average number of generations, and convergence behaviors of average fitness values have been measured in each algorithm. The computational results have shown that the proposed HGA with local search technique is more efficient in most of the performance comparisons than the BGA. Especially, the latter has good convergence.

For future study, we have a plan to develop various techniques that can be used in GA strategy parameters such as population size and crossover rate.

References

[1] Kim, K.W., Yun, Y.S., Yoon, J. M., Gen, M. & Yamazaki, G. Hybrid genetic algorithm with adaptive abilities for resource constrained multiple project scheduling, Computers in Industry, 2005, 56(2), 143–160.

[2] HOLLAND JH. Adaptation in Natural and Artificial System[M]. Ann Arbor: University of Michigan Press, 1975.

[3] Mingwang Zhao. A Hybrid Numerical Algorithm for Function Optimization Based on Genetic Algorithm and Steepest Decent Algorithm [J]. System Engineering Practice and Application. 1997(7), 60-63.

[4] Peizhi Li, Ding Fan. Improved Genetic Algorithm Based Real Coding [J]. Journal of Astronautic Metrology and Measurement. 2006(8), 1960-1965.

[5] Lin Zhang, Zhong Zheng, Xiaoqiang Gao. Hybrid Genetic Algorithms for Multi-modal Optimization [J]. Chongqing University Natural Journal. 2005, 28(7): 51-54.

[6] Michalewicz,Z.(2004). Genetic algorithms + data structures = evolution program. Berlin:Spring-Verlag. Rabi, V., Murty, B. S. N., &Reddy, P. J. (1997). Non-equilibrium simulated annealing algorithm applied to reliability optimization of complex systems [J]. IEEE Transactionson Reliability, 46(2),233-239.

[7] Park CH, Lee WI, Han WS, VautrinA. Simultaneous optimization of composite Structures considering mechanical performance and manufacturing cost. Composite Struct [J] 2004, 65 (1):117-127.

[8] Saeid Fallah-Jamshidi, Maghsoud Amiri, Neda Karimi. Nonlinear continuous multi-response problems:a novel two-phase hybrid genetic based metaheuristic [J]. Applied Soft Computing. 2010(10):1274-1283.

Table 1 Constant coefficients for problem 2

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r_j	.90	.75	.65	.80	.85	.93	.78	.66	.78	.91	.79	.77	.67	.79	.67
c_j	5	4	9	7	7	5	6	9	4	5	6	7	9	8	6
w_j	8	9	6	7	8	8	9	6	7	8	9	7	6	5	7

Table 2 Computational results of BGA and HGA

Problem 2		
	BGA	HGA
BFV	0.7417	0.0000
AFV	4.3884	1.0012
NGS	20	13
ANG	5000	2033
CPU(sec.)	5.04	3.75