http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (3): 524-528, 2006
ISSN 1812-5638

Adaptive SAGA Based on Mutative Scale Chaos Optimization Strategy

[1]Haichang Gao, [1]Boqin Feng, [1]Yun Hou, [2]Bin Guo and [3]Li Zhu
[1]Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China
[2]Anzai and Imai Laboratory, Keio University, Yokohama 223-8522, Japan
[3]School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Abstract: A hybrid adaptive SAGA based on mutative scale chaos optimization strategy (CASAGA) is proposed to solve the slow convergence, incident getting into local optimum characteristics of the Standard Genetic Algorithm (SGA). The algorithm combined the parallel searching structure of Genetic Algorithm (GA) with the probabilistic jumping property of Simulated Annealing (SA), also used adaptive crossover and mutation operators. The mutative scale Chaos optimization strategy was used to accelerate the optimum seeking. Compared with SGA and MSCGA on some complex function optimization and several TSP combination optimization problems, the CASAGA improved the global convergence ability and enhanced the capability of breaking away from local optimal solution.

Key words: Genetic algorithm, simulated annealing, chaos, TSP, optimization

INTRODUCTION

Genetic Algorithm (GA) is a general stochastic optimization strategy based on the principle of biologic evolution and natural selection. The basic concepts of GA were developed by Holland (1973) who inspired by Darwin's the survival of the fittest theory. GA includes a class of adaptive searching techniques that are suitable for searching a discontinuous space. The elementary operations are reproduction, crossover and mutation. Genetic algorithms maintain a population of solutions rather than just one current solution. The population is iteratively recombined and mutated to generate successive populations. There are implicit parallelism search characteristics in GA, it can deal with some complex optimization problem that can't do by traditional methods (Eiben *et al.*, 1999). But GA has some disadvantages in dealing with combined optimum problem for complexes structure with large search space, long search time and premature convergence.

SA is based on the idea of neighborhood search. Kirkpatrick *et al.* (1983) suggested a form of SA could be used to solve complex optimization problems. The algorithm works by selecting candidate solutions which are in the neighborhood of the given candidate solution. SA attempts to avoid entrapment in a local optimum by sometimes accepting a move that deteriorates the value of the objective function (Ahmed and Alkhamis, 2002). With the help of the distribution scheme, SA can provide a reasonable control over the initial temperature and cooling schedule so that it performs effective exploration and good confidence in the solution quality.

Chaos movement can go through all states unrepeatedly according to the rule of itself in some area. It was introduced into the optimization strategy to accelerate the optimum seeking operation (Yan *et al.*, 2002).

In this paper, combining the parallel search ability of a kind of adaptive GA with the controllable jumping property of SA (Ling and Dazhong, 2003) and combined using mutative scale chaos strategy, a kind of CASAGA hybrid meta-heuristic algorithm with the operators and parameters reasonably designed is proposed. The experiments on some benchmark programs have shown that the CASAGA hybrid meta-heuristic algorithm may significantly reduce the cost and decrease the probability of getting into local optimum comparing to SGA and MSCGA.

MUTATIVE SCALE CHAOS OPTIMIZATION STRATEGY

Chaos is one of the most popular phenomenons that exist in nonlinear system and theory of chaos is one of the most important achievements of nonlinear system research. It is now widely recognized that chaos is a fundamental mode of motion underlying almost natural phenomena (Bing and Weisun, 1997).

Corresponding Author: Haichang Gao, Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China

Logistic equation is brought forward for description of the evolution of biologic populations (Moon, 1992). It is the most common and simple chaotic function:

$$x_{n+1} = L\ x_n(1 - x_n) \quad (1)$$

Where, L is a control parameter which is between 0 and 4.0. When L = 4.0, the system is proved to be in chaotic state. Given arbitrary initial value that is in (0,1) but not equal with 0.25, 0.5 and 0.75, chaos trajectory will finally search non-repeatedly any point in (0,1).

If the target function of continuous object problem that to be optimized is:

$$f^3 = f(x_i^3) = \min f(x_i),\ x_i \quad [a_i, b_i],\ i = 1, 2, \quad, n \quad (2)$$

Then the basic process of the mutative scale chaos optimization strategy can be described as follows:

Step 1, algorithm initialization. Let k = k' = 1, $x_i(k)$ - $x_i(0)$, $x_i^3 = x_i(0)$, $f^3 = f(0)$, $a_i(k') = a_i$, $b_i(k') = b_i$. where, k is the iterative symbol of chaos parameters. k' is the refine search symbol. x_i^3 is the best chaos variable found currently. f^3 is the current best solution that initialized as a biggish number.

Step 2, map the chaos variable $x_i(k)$ to the optimization variable area, get $mx_i(k)$:

$$mx_i(k) = a_i(k') + x_i(k)(b_i(k') - a_i(k')) \quad (3)$$

Step 3, search according to the chaos optimization strategy. $f^3 = f(mx_i(k))$, $x_i^3 = x_i(k)$, if $f(mx_i(k)) < f^3$. Otherwise, go on.

Step 4, let k = k+1, $x_i(k) = 4x_i(k)(1-x_i(k))$, repeat step 2 and 3 until f^3 keep unchanged in certain steps.

Step 5, reduce the search scale of chaos variable:

$$a_i(k'+1) = mx_i^3 - C(b_i(k') - a_i(k')),$$

$$b_i(k'+1) = mx_i^3 + C(b_i(k') - a_i(k')) \quad (4)$$

Where, adjustment coefficient $C \in (0, 0.5)$, m x_i^3 is the best solution currently.

Step 6, revert optimization variable x_i^3:

$$x_i^3 = \frac{mx_i^3 - a_i(k'+1)}{b_i(k'+1) - a_i(k'+1)} \quad (5)$$

Repeat step 2 to 5 using new chaos variable $y_i(k) = (1-A)x_i^3 + A\,x_i(k)$, where A is a small number. Let k' = k'+1, until f^3 keep unchanged in certain steps.

Step 7, finish the calculate process after several repeating of step 5 and 6. The final m x_i^3 is the best optimization variable and f^3 is the best solution.

ADAPTIVE SAGA BASED ON CHAOS STRATEGY

There exist some weaknesses in simple GA and SA. SA has a continuous characteristic, so real-designed SA often suffers from the difficulty of proper control over the process and prohibitive time-consumption required for equilibrium. GA presents implicit parallelism and can retain useful redundant information about what is learned from previous searches by its representation in individuals in the population. But GA may lose solutions and substructures due to the disruptive effects of genetic operators and it is not easy to be premature and results in poor solutions. Based on the suitable cooling schedule, SA has good convergence property and the ability to probabilistically escape from local optima can be controlled.

Chaos has three important dynamic properties: the sensitive dependence on initial conditions, the intrinsic stochastic property and ergodicity. Chaos is in essence deeply related with evolution. In chaos theory, biologic evolution is regarded as feedback randomness, while this randomness is not caused by outside disturbance but intrinsic element (Tong *et al*., 1999). So it is believed that chaos is the source of information and system evolution. Taking advantage of chaos, a new search algorithm called chaotic search is presented which has the better capacity of climbing hill comparing with simulated annealing and random search. But in must almost search the whole solution space using an intrinsic stochastic property to get optimum solution, which needs more computation time, another shortage is that chaos search operates only one to one.

Thus, a hybrid meta-heuristic algorithm combined of GA and SA based on chaos optimization strategy is presented as follows:

Algorithm CASAGA:

Step 1, Initialization. Set population size M, crossover probability P_c, mutation probability P_m, initial temperature T_0, cooling parameter α and maximum generation Gen;

Step 2, Generate initial population $\{x_1, x_2, \Lambda, x_m\}$ in enactment range;

Step 3, Calculate fitness function $f(x_1)$ of each $x_1 (i = 1,2,\ldots,m)$;

Step 4, Produce the next new higher survivor generation according to the fitness function (Roulette selection);

Step 5, Perform crossover and mutation operation with corresponding probability P_c and P_m(formula (6) and (7));

Step 6, Carry chaos search to the individuals with high fitness value using mutative scale chaos strategy, lead the population rapid evolution;

Step 7, Perform annealing operation, Decrease temperature $t_{k+1} = \text{update}(t_k)$ and set k = k+1;

Step 8, Terminate the algorithm if the stopping criteria is reached and output the final results. Otherwise, goto Step 3.

End.

During the hybrid search process, GA provides a set of initial solutions for SA at each temperature to perform Metropolis sample for each solution until equilibrium condition is reached and GA uses the solutions found by SA to continue parallel evolution. The optimization operators, such as mutation operator and the new solution generator of SA, can be different or hybrid used to yield a large neighborhood and efficiently explore better solutions among the solution space.

Fitness function and annealing function: The fitness function was used to measure the fitness of the individual; it's the only joint of meta-heuristic algorithm and automatic software test case generation.

In Annealing function construction, exponential cooling schedule is used to adjust the temperature $t_{k+1} = \omega \cdot t_k$, where, $\omega \in (0,1)$ is a decrease rate. It is often believed to be an excellent cooling method, because it provides a rather good compromise between a computationally fast schedule and the ability to reach low-energy state.

Selection strategy: The main selection operators are tournament selection and roulette selection. The roulette selection method is selected in the paper. Each individual corresponds to one segment in roulette according to its fitness value percentage. Then the wheel needs to roll for N times to pick N parents. The marked position in wheel being selected becomes next parents after the rolling.

Crossover strategy: The arithmetic crossover operation

$$\begin{aligned} x_1' &= P_c\ x_1 + (1-P_c)\ x_2 \\ x_2' &= P_c\ x_2 + (1-P_c)\ x_1 \end{aligned},$$

is implemented for the selected two solutions, where, x_1 and x'_2 are parents, $x'_1$and x_2 are children, $P_c \in (0,1)$is a random variable. Such procedure is repeated M/2 times to generate the new population, then the top M solutions with better objective values from the old population and new solutions are reserved (Srnivas and Patnaik, 1994). The crossover probability P_c can be calculated using the following equation:

$$P_c = \begin{cases} \dfrac{\mathrm{sqrt}(f_{max}^2 - f_c^2)}{f_{max} - f_{avg}}, & f_c \ge f_{avg} \\ 1 & f_c < f_{avg} \end{cases} \tag{6}$$

Where, f_{max} denotes the maximum fitness value of the current generation; f_{avg} denotes the average fitness value; f_c denotes the fitness value of the chromosome which carry out the crossover operation.

Mutation strategy: By incorporating SA into the search structure, SA serves a type of mutation with adaptive probability controlled by temperature. So SA can be regarded as a type of mutation operation with adaptive rate to avoid being trapped in certain solutions at a high temperature and serves as a fine neighbor search at a low temperature. Thus, mutation rate is set to one to perform a fine chemotactic neighbor search and all these operators are designed as local search operators, which can be conducted by appending random noise for each parameter $w' = w + \eta \cdot \xi$, where, ξ is a random variable subjected to Gaussian distribution N(0,1) and η is a scale parameter. The mutation probability P_m can be calculated using the following equation:

$$P_m = \begin{cases} \dfrac{\mathrm{sqrt}(f_{max}^2 - f_m^2)}{2(f_{max} - f_{avg})}, & f_m \ge f_{avg} \\ 1/2 & f_m < f_{avg} \end{cases} \tag{7}$$

Where, f_m denotes the fitness value of the chromosome which carry out the mutation operation.

CONVERGENCE ANALYSIS

The hybrid algorithm CASAGA that presented based on the chaos optimization can reduce the probability of getting into the local optimum and it can guarantee to convergence to overall optimum.

According to literature of Yadong and Shaoyuan (2002), let x^3 = arg min f(x) for the logistic map is used in the optimization search. $\{x_1^k\}$ is the solution list produced by CASAGA, $\{x_2^k\}$ is the solution list produced by annealing genetic algorithm. For $i \le j$, we can get $f(x_1^i) \ge f(x_1^j)$, $f(x_2^i) \ge f(x_2^j)$. For $\{f(x_2^k)\}$ is a convergence list and lim $P\{f(x_2^k = f(x^3\} = 1$, $f(x_1^k) \le f(x_2^k)$, $f(x_2^k) \ge f(x_1^k) \ge f(x^3)$, so $f(x_1^k)$ is also a convergence list according to converging and approximating theorem.

RESULTS AND DISCUSSION

Complex functions: Some benchmark functions (F_1 to F_3) were carried to compare CASAGA with SGA and MSCGA (Goldberg, 1989). These functions get into local optimum easily. There are numerous local optima around the global optimum (Fig. 1 and 2).

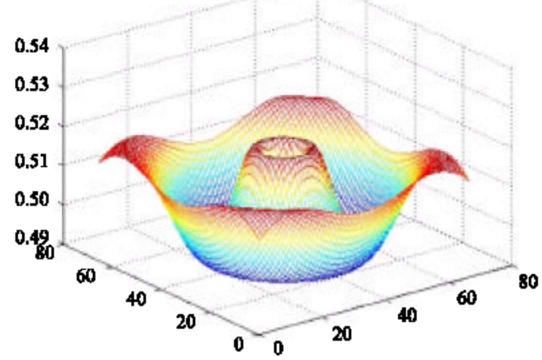

Fig. 1: Figures of complex testing function F_1

Table 1: A comparison of three algorithm on benchmark functions

		F1	F2	F3
Optimized point		(0,0,0)	(1,1)	(0,0)
Optimized value		0	0	1
Percentage of convergence (%)	SGA	53	71	34
	MSCGA	96	100	100
	CASAGA	100	100	100
Feasible solution obtained	SGA	3562	4003	737
	MSCGA	1125	2074	986
	CASAGA	958	1665	581
Time spend in finding optimum (S)	SGA	0.1512	0.2541	5.2586
	MSCGA	0.2098	0.4617	11.2551
	CASAGA	0.3466	0.5788	13.1149

Table 2: Experimental results on three symmetry TSP problems

	Best solution	Algorithm	Average solution	Average time (s)	Average generation
Burma14	3323	CASAGA	3323.1	15.5	37
		MSCGA	3324.5	14.4	59
		SGA	3332.7	6.5	64
Att48	10628	CASAGA	10629.3	43.5	38
		MSCGA	10636.9	39.1	57
		SGA	10648.1	28.4	86
Ch130	6110	CASAGA	6110.8	37.7	47
		MSCGA	6113.7	23.2	68
		SGA	6119.6	16.5	91

Table 3: Experimental results on three asymmetry TSP problems

	Best solution	Algorithm	Average solution	Average time (s)	Average generation
Ftv33	1286	CASAGA	1286.0	25.8	16
		MSCGA	1286.3	23.1	27
		SGA	1287.1	8.4	33
Ft53	6905	CASAGA	6905.2	52.9	38
		MSCGA	6907.4	46.4	46
		SGA	6909.8	28.9	52
Rbg403	2465	CASAGA	2465.7	66.7	57
		MSCGA	2467.4	53.6	86
		SGA	2470.3	30.5	105

$$F_1 = 0.5 - \frac{\sin\sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}, \ -100 \le x_i \le 100 \quad (8)$$

$$F_2 = 100(x_1 - x_2) + (1 - x_1)^2, -2.048 \le x_i \le 2.048 \quad (9)$$

$$F_3 = \sum_{i=1}^{3} x_i^2, -5.12 \le x_i \le 5.12 \quad (10)$$

CASAGA outperforms SGA and MSCGA in the mass. The CASAGA can convergent to global optimum 100 percent, but SGA can't. From the result of feasible solutions, we can see CASAGA is superior to other two algorithms. It proved the CASAGA that presented in the study is efficient and promising (Table 1).

TSP problem: The travelling salesman problem (TSP) has been proved as a NP-Hard problem (Garey and Johnson, 1979). It represent a kind of combination optimization problem and has a lot of application in practical engineering. It has been used to study the performance of

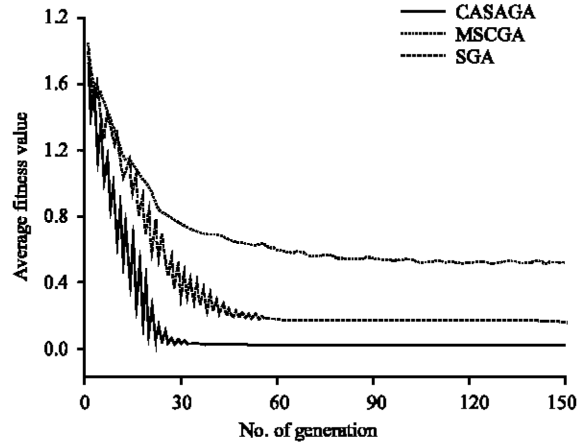

Fig. 2: Average fitness value curve of function F_1

algorithm according to its important engineering and academic value.

TSP is the problem of finding a shortest closed tour which visits all the cities in a given set. Its mathematical description is follow: Given a cities set $C = (c_1, c_2, ... c_n)$, where, distance of every pair cities is $d(c_i, c_j) \in R^+$ The problem is finding a tour $(c_{\pi 1}, c_{\pi 2}, ... c_{\pi n})$ visits all the cities once and makes

$$\min \sum_{i=1}^{n-1} d\left(c_{\pi i}, c_{\pi(i+1)}\right) + d\left(c_{\pi n}, c_{\pi 1}\right) \quad (11)$$

Where, $\pi 1, \pi 2, ... \pi n$ is the permutation of (1,2,...,n).

Several symmetry and asymmetry TSP prolems from TSP general standard library TSPLIB (TSPLIB, 2005) was used to validate the validity of the algorithm. Generation size is 50, the maximum iterative number is 500. The crossover and mutate probability of CASAGA changs adaptively. The initial temperature of annealing process is 2000, the end temperature is 0 and the annealing speed is 0.98. The results of experiment has been shown in Table 2 and 3.

The average solutions of CASAGA is close to the known best solution and need less iteration on the symmetry and asymmetry TSP prolems. One disadvantaeg is that it needs more search time for the annealing operation was introduced into each individual. But the damnify is acceptable relative to the improvement of the performance. Above experiments were carried on Pentium4-2.4G/512M PC platform with MATLAB Tools.

CONCLUSIONS

This study discusses a kind of CASAGA hybrid meta-heuristic algorithm which combines the characteristics of GA, SA and Chaos.

Take no account of chaos search, the hybrid algorithm combines SA and GA through SA-Mutation (SAM) and SA-Recombination (SAR) operations. The algorithm translates into standard GA if the probability of SAM and SAR is zero. Chaos movement can go through all states unrepeated according to the rule of itself in some area. It was used to seek for optimum in population which was operated by SAGA. It can guide the population evolve rapidly.

Experimental results based on some benchmark functions show that CASAGA is quite flexible with satisfactory results than SGA and MSCGA. It's an efficient and promising optimization strategy.

ACKONWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful and constructive comments. This work was supported in part by the National High Technology Development Plan of China (863) under grant no. 2003AA1Z2610.

REFERENCES

Ahmed, M.A. and T.M. Alkhamis, 2002. Simulation based optimization using simulated annealing with ranking and selection. Comp. Operat. Res., 29: 387-402.

Bing, L. and J. Weisun, 1997. Chaos optimization method and its application. Control Theory and Applications, 14: 613-615.

Eiben, A.E., R. Hinterding and Z. Michalewicz, 1999. Parameter control in evolutionary algorithms. IEEE Trans. Evot. Computation, 3: 124-141.

Garey, M.R. and D.S. Johnson, 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: Freeman WH.

Goldberg, D.E., 1989. Genetic algorithm in search, optimization and machine learning. MA: Addison Wesley Reading, pp: 1-10.

Holland, J.H., 1973. Genetic algorithms and the optimal allocation of trials. SIAM J. Comp., pp: 89-104.

Kirkpatrick, S., J.C.D. Gelat and M.P. Vecchi, 1983. Optimization by simulated annealing. Sci., 220: 671-680.

Ling, W. and Z. Dazhong, 2003. An effective hybrid heuristic for flow shop scheduling. Intl. J. Adv. Manufact. Technol., 21: 38-44.

Moon, F.C., 1992. Chaotic and Fractal Dynamics, an Introduction for Applied Scientists and Engineers. New York: John Wiley and Sons.

Srnivas, M. and L.M. Patnaik, 1994. Adaptive probability of crossover and mutation in genetic algorithms. IEEE Trans. SMC, 24: 656-667.

Tong, Z., W. Hongwei and W. Zicai, 1999. Mutative scale chaos optimization algorithm and its application. Control and Decision, 14: 285-288.

TSPLIB, 2005. http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/index.html.

Yadong, L. and L. Shaoyuan, 2002. A new genetic chaos optimization combination method. Control Theory and Applications, 19: 143-145.

Yan, W., L. Jinlu and S. Yikang, 2002. Hybrid genetic algorithm based on mutative scale chaos optimization strategy. Control and Decision, 17: 958-960