SOMGA for Large Scale Function Optimization and Its Application

Dipti Singh and Kusum Deep

Abstract Self Organizing Migrating Genetic Algorithm (SOMGA) is a hybridized variant of Genetic Algorithm (GA) inspired by the features of Self Organizing Migrating Algorithm, presented by Deep and Dipti (IEEE Congr Evol Comput, pp 2796–2803, 2007) [1]. SOMGA extracts the features of binary coded GA and real coded SOMA in such a way that diversity of the solution space can be maintained and thoroughly exploited keeping function evaluation low. It works with very less population size and tries to achieve global optimal solution faster in less number of function evaluations. Earlier SOMGA has been used to solve problems up to 10 dimensions with population size 10 only. This chapter is brake into three sections. In first section a possibility of using SOMGA to solve large scale problem (dimension up to 200) has been analyzed with the help of 13 test problems. The reason behind extension is that SOMGA works with very small population size and to solve large scale problems (dimension 200) only 20 population size is required. On the basis of results it has been concluded that SOMGA is efficient to solve large scale global optimization problems with small population size and hence required lesser function evaluations. In second section, two real life problems from the field of engineering as an application have been solved using SOMGA. In third section, a comparison between two ways of hybridization has been analyzed. There can be two approaches to hybridize a population based technique. Either by incorporating a deterministic local search in it or by merging it with other population based technique. To see the effect of both the approaches on GA, the results of SOMGA on five test problems are compared with the results of MA (GA+ deterministic local search). Results clearly indicates that SOMGA is less expensive and effective to solve these problems.

D. Singh (✉)
Department of Applied Sciences, Gautam Buddha University, Greater Noida, India
e-mail: diptipma@rediffmail.com

K. Deep
Indian Institute of Technology Roorkee, Roorkee, India
e-mail: kusumfma@iitr.ac.in

Keywords Self organizing migrating algorithm · Genetic algorithm · Large scale global optimization · Real life problems

1 Introduction

A variety of computational techniques have appeared in literature for solving nonlinear optimization problems. However, there is no single technique that can claim to efficiently solve each and every nonlinear optimization problem. In fact, a technique, which is efficient for one type of nonlinear optimization problem, may be inefficient for solving other types of nonlinear optimization problems. Moreover, the computational time and memory space requirements of most of these techniques are quite large. Keeping this in view, it is desirable to develop more efficient and reliable computational techniques for solving a large variety of nonlinear real life optimization problems of practical interest which require less memory space so that these can be easily implemented on personal computers. Some of the general requirements of a computational algorithm for solving global optimization problems are: (i) wide applicability i.e., it should be applicable to a wide class of real life problems (ii) simplicity in structure, which would allow it to be easily implemented on a computer system (iii) minimum mathematical complexities, so that it can be used conveniently, even by non expert users.

Optimization problems arise in several fields such as system engineering, telecommunication and manufacturing systems etc. In fact the newly developed optimization techniques are now being extensively used in various spheres of human activity where decisions have to be taken in some complex situations that can be represented by mathematical models. A real life optimization problem may have a number of local as well as global optimal solutions. It is desired by most of the users to design optimization techniques which determine the global optimal solution rather than the local optimal solution of nonlinear optimization problems. With the advent of computers, population based heuristics are becoming popular day by day, not only because of their ease of implementation, but also due to their wide applicability. Population based stochastic search methods have been frequently used in the literature to solve real life global optimization problems. Although these probabilistic search algorithms do not give absolute guarantee to determine the global optimum solution, these methods are preferred over traditional methods. Evolutionary Algorithms, particularly Genetic Algorithms (GAs) are the most commonly used heuristics for solving real life problems. Though GAs are efficient to solve global optimization problem but usually converges very slow and generally required large population size also. Many attempts have been made in literature to improve the efficiency of Genetic algorithms either by designing new operators or by incorporating the features of other techniques. Grefensette [2] introduced a hybrid variant of GA which uses a traditional hill climbing routine for improving the fitness of newly generated points known as Fawwin and Lamarckian evolution approach and then the new offspring compete with their parents for selection as a member of the next population. Kasprzyk and Jasku [3] developed a variant of GA which is hybridization of GA and simplex method known as genetic—simplex algorithm, in this approach first the solution area is explored by GA operators and then simplex method uses starting points provided by the GA to determine an optimum solution. Chelouah and Siarry [4] also proposed a hybrid method by combining the features of continuous Tabu search and Nelder-Mead simplex algorithm. This approach is used to find the global minima of nonlinear optimization problems. Wang et al. [5] apply quantum computing to GAs to develop a class of quantum-inspired GAs.

Javadi et al. [6] presented a neural-network based genetic algorithm which uses neural network to improve solution quality and convergence speed of GAs. Fan et al. [7] integrate the Nelder–Mead Simplex search method with genetic algorithm and particle swarm optimization in an attempt to locate the global optimal solution of nonlinear continuous variable functions focusing mainly on response surface methodology. Comparative performance on ten test problems is demonstrated. Hwang and Song [8] present a novel adaptive real-parameter simulated annealing genetic algorithm which maintain the merits of GAs and simulated annealing. Zhang and Lu [9] define a new real valued mutation operator and use it to design a hybrid real coded GA with quasi-simplex technique. A nitche hybrid genetic algorithm is proposed by Wei and Zhao [10] and results are reported on 3 benchmark functions. Premalatha and Nataranjan [11] established hybrid PSO which proposes the modification strategies in PSO using GA to solve the optimization problems. Khosravi et al. [12] proposes a novel hybrid algorithm that uses the abilities of evolutionary and conventional algorithm simultaneously. Ghatei et al. [13] designed a new variant of particle swarm optimization by including Great Deluge Algorithm (GDA) as local search factor. Esmin and Matwin [14] presented a hybrid approach of using the features of PSO and GA known as HPSOM algorithm. The main idea of this approach is to integrate the PSO with genetic algorithm mutation method.

It is clear from the literature that several variants of GA are available in literature to improve the efficiency of these algorithms. Deep and Dipti [1] presented a variant of GA named as SOMGA in which GA has been hybridized with a new emerging population based technique, Self Organizing Migrating Algorithm (SOMA). SOMA is an emergent search technique in the field of population based techniques, developed by Zelinka and Lampinen [15]. This algorithm is based on the self organizing behavior of group of individuals looking for food. For this all individuals follow the path of one individual known as Leader selected among them based on the best fitness value. In the whole process of this algorithm, no new solutions are created during the search. Instead, only the positions of the solutions are changed during a generation, called a migration loop and it works with small population size. The details of this algorithm can be found in many research papers and books Oplatkova and Zelinka [16], Zelinka [17], Nolle and Zelinka [18], Nolle et al. [19], Nolle [20], Zelinka et al. [21, 22], Onwubolu and Babu [23], etc. The common feature between GA and SOMA is that both are population based stochastic search heuristics. Mutation and crossover is done (but the way in which they are applied is different). Some differences of the two algorithms are as follows:

- In GA new points are generated, whereas in SOMA no new points are generated, but instead their positions are updated.
- Individuals can proceed in any direction in GA, whereas in SOMA individuals proceed only in the direction of the leader.
- GA has a competitive behavior, but SOMA has competitive-cooperative behavior.
- GA works with a large population size, whereas SOMA works with a small population size.

2 Previous Work Done

The algorithm SOMGA has been designed to solve the unconstrained non-linear optimization problems of the type:

$$\left.\begin{array}{r} \operatorname{Min} f(X), X = (x_1, x_2, \ldots x_n) \\ a_i \le x_i \le b_i, i = 1, 2, \ldots n. \end{array}\right\} \qquad (1)$$

where a_i and b_i are the lower and upper bounds on the variables.

As discussed above several variants of population based techniques are available in literature, for improving the convergence of these algorithms. The main reason of slow convergence and premature convergence of these algorithms is considered as diversity mechanism. If one algorithm fails to maintain the diversity during the search then there are more chances to be converging premature. SOMGA is an effort to improve the efficiency of both the algorithms GA and SOMA by extracting the best features of these algorithms. In the hybridization of SOMGA, binary coded GA and real coded SOMA has been used. It derives the features of selection, crossover and mutation from binary coded GA and derives the features of small population size, organization and migration from real coded SOMA. The features of GA and SOMA are combined in such a way that solution search space can be thoroughly exploited and diversity of the search domain can be preserved by generating new points in solution space. The methodology of this algorithm is:

Methodology

First the individuals are generated randomly. These individuals compete with each other through tournament selection; create new individuals via single point crossover and bitwise mutation. Then the best individual among them is considered as leader and the worst individual is considered as active. The active individual proceeds in the direction of the leader in n steps of the defined length. This path is perturbed randomly by a parameter known as PRT parameter. It is defined in the range $\langle 0, 1 \rangle$. Using this PRT parameter value, PRT vector is created before an individual proceeds towards leader. This parameter has the same effect as mutation in GA. The movement of an individual is given as follows:

$$x_{i,j}^{MLnew} = x_{i,j,start}^{ML} + \left(x_{L,j}^{ML} - x_{i,j,start}^{ML} \right) tPRTVector_j \tag{2}$$

where $t \in \langle 0, by\ Step\ to, PathLength \rangle$,

ML	is actual migration loop.
$x_{i,j}^{MLnew}$	is the new positions of an individual.
$x_{i,j,start}^{ML}$	is the positions of active individual.
$x_{L,j}^{ML}$	is the positions of leader.

At last the best individuals (number equal to population size) from the previous and current generations are selected for the next generation. The computational steps of this approach are given below:

Step 1: Generate the initial population.
Step 2: Evaluate all individuals
Step 3: Apply tournament selection on all individuals to select the better individuals for the next generation.
Step 4: Apply crossover operator on all individuals with crossover probability P_c to produce new individuals.
Step 5: Evaluate the new individuals.
Step 6: Apply mutation operator on every bit of every individual of the population with mutation probability P_m.
Step 7: Evaluate the mutated individuals.
Step 8: Find leader (best fitted individual) and active (worst fitted individual) of the population.
Step 9: For active individual a new population of size N is created. where N = (Path Length/step size). This population is nothing but the new positions of the active individual towards the leader in n steps of the defined length. The movement of this individual is given in Eq. (2).
Step 9.1: Select the best individual of the new population and replace the active individual with this best individual.
Step 10: Select the best individuals (in fitness) of previous and current generation for the next generation via tournament selection.
Step 11: If termination criterion is satisfied stop else go to Step 3.
Step 12: Report the best chromosome as the final optimal solution.

Salient Features of SOMGA

The salient features of the SOMGA are:

1. SOMGA attempts to determine the global optimal solution of nonlinear unconstrained optimization problems.
2. SOMGA does not require the continuity and/or differentiability conditions of the objective function.

3. SOMGA works on purely function evaluations and hence can be used in situations where the objective function is discontinuous in nature.
4. SOMGA does not require an initial guess value to start, but instead SOMGA requires a lower and upper bound for the unknown variables.

3 Section 1

3.1 Solution of Large Scale Problems Using SOMGA

Now a days, to solve large scale optimization problems using evolutionary algorithms has become the new field of research. In large scale problems, where the number of unknown variables vary up to 200 not only the computational time but also the memory space requirements becomes very large. Besides this the complexity of the problem also increases significantly that many algorithms fails to reach the optimal solution. One algorithm able to reach optimal solution requires large population size as well as functional evaluations. The possibility of using SOMGA for solving large scale problems is considered in this section. One possible way to reduce the computational time and also to reduce the memory space is to reduce the population size. Since SOMGA already works with less population size hence there is no need of reducing population size. Reduction in memory space is also not required because the population generated for the active individuals takes memory space only for short time. After choosing the best one of the generated population, active individual gets replaced by this best individual and other individuals release the memory space. So here the chances for using it to solve large scale problems are very strong. The problems up to 200 variables can be solved on Pentium IV normally configured system.

3.2 Results and Discussion

In order to observe the performance of SOMGA on large scale problems a set of thirteen test problems have been selected given in appendix. These problems are scalable in nature that is their size can be increased or decreased as per the user's choice. In general, the complexity of the problem increases as the dimension of the problem increases. Here in Griewank function the complexity at dimension 10 is maximum. The complexity of this function decreases as the dimension increases above 10.

The parameters of SOMGA used to solve 100 and 200 dimension are same and are given in Table 1. These parameters are population size, crossover rate, mutation rate, string length, PRT, step size, path length and total number of function calls

Table 1 Parameters of SOMGA for problem size 100 and 200

Population size	20
Crossover rate	0.95
Mutation rate	0.0001
String length	30
PRT	1
Step	0.091
Path length	3
Total number of function calls allowed	350,000

Table 2 Results of large scale problems for problem size 10

Problem	Mean	Standard deviation	% Success	Function evaluations
Cosine mixture function	0.996	0.003	100	4336
Exponential function	0.994	0.003	100	2028
Ackley function	0.007	0.002	100	9644
Sphere function	0.007	0.003	100	3790
Griewank function	0.068	0.036	20	13,639
Axis parallel hyper ellipsoid function	0.006	0.003	100	4137
Schwefel's double sum function	0.007	0.003	100	7324
Restrigin's function	0.003	0.003	100	16,895
Rosenbrock function	4.699	1.889	0	NA
Schwefel's function	0.004	0.003	100	11,660
Zakhrov's function	0.005	0.003	100	4939
Ellipsoidal function	0.006	0.003	100	3468
Schwefel's problem 4 function	0.005	0.003	100	310

allowed. Main thing to notice in Table 1 is that for solving 100 and 200 dimensional problems, only 20 population size is required. The results for problem size 10 are given in Table 2 taken from Deep and Dipti [1].

Here, first we apply SOMGA for problem size 100. The percentage of success, average function evaluations, mean and standard deviation of the objective function values of 30 runs are recorded in Table 3. The same information for problem size 200 is shown in Table 4. These values are also shown graphically in Figs. 1 and 2. It can be observed in Table 3 that SOMGA gives 100 % success in 11 problems out of thirteen. In Griewank problem the success rate is 40 % and Rosenbrock could not be solved at all.

In Table 4, the results for a problem size of 200 are presented. It is notable that SOMGA gives 100 % success in 10 problems out of thirteen. In Griewank problem,

Table 3 Results of large scale problems for problem size 100

Problem	Mean	Standard deviation	% Success	Function evaluations
Cosine mixture function	−9.99544	0.003721	100	66,865
Exponential function	−0.99245	0.002478	100	35,523
Ackley function	0.008684	0.00129	100	126,121
Sphere function	0.0077	0.002573	100	75,449
Griewank function	0.050896	0.071455	40	259,559
Axis parallel hyper ellipsoid function	0.008083	0.001724	100	85,889
Schwefel's double sum function	0.006285	0.002476	100	126,078
Restrigin's function	0.004288	0.004288	100	185,626
Rosenbrock function	95.24293	1.177368	0	NA
Schwefel's function	0.009224	0.000703	100	248,930
Zakhrov's function	0.006024	0.002964	100	156,021
Ellipsoidal function	0.008618	0.002317	100	118,629
Schwefel's problem 4 function	0.004494	0.002769	100	521

Table 4 Results of large scale problems for problem size 200

Problem	Mean	Standard deviation	% Success	Function evaluations
Cosine mixture function	−19.9926	0.003517	100	114,165
Exponential function	−0.99016	0.00015	100	61,248
Ackley function	0.008969	0.001514	100	203,392
Sphere function	0.009564	0.000559	100	115,071
Griewank function	0.05995	0.109189	60	257,840
Axis parallel hyper ellipsoid function	0.00962	0.000466	100	180,245
Schwefel's double sum function	0.008717	0.002257	100	246,508
Restrigin's function	0.005262	0.004334	100	272,246
Rosenbrock function	221.4295	35.95709	0	NA
Schwefel's function	4159.555	166.9408	0	NA
Zakhrov's function	0.009576	0.000521	100	255,085
Ellipsoidal function	0.008911	0.001696	100	311,626
Schwefel's problem 4 function	0.004653	0.003564	100	527

the success rate is 60 % for problem size 200 where as for problem size 100 it is 40 %. The success rate in Griewank problem increases as the dimension of the problem increases. This shows the behavior of Griewank problem on increasing the

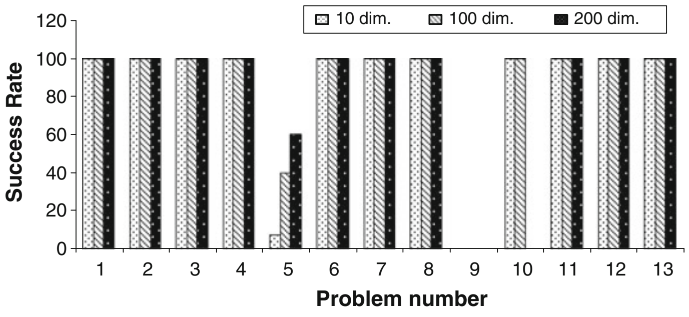

Fig. 1 Performance of SOMGA in terms of success for 100 dimension and 200 dimension problems

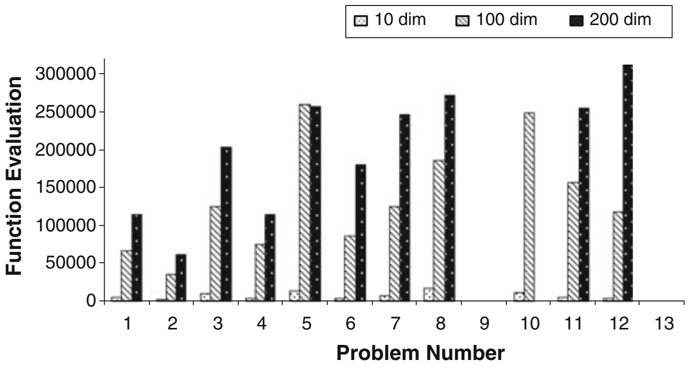

Fig. 2 Performance of SOMGA in terms of function evaluations for 100 dimension and 200 dimension problems

dimension that is, the complexity of the Griewank problem decreases as the dimension increases. In Rosenbrock problem SOMGA fails. Schwefel problem gives 100 % success for problem size 100, but could not be solved at all for problem size 200. This shows that the complexity of the schewefel problem increases as the dimension increases.

In Fig. 1, the success rate obtained by SOMGA in 10 dimension, 100 dimension and 200 dimension problems is plotted. Since the complexity of the Griewank function decreases as the dimension of the problem increases. Same kind of behavior can be seen in Fig. 2 problem number 5. In Fig. 2, the function evaluations required by SOMGA in 10 dimension, 100 dimension and 200 dimension problems are plotted. The behavior is obvious. Function evaluations are increasing as the dimension of the problem is increasing. In Griewank problem the function evaluations in 100 dimesion and 200 dimensions are almost same.

4 Section 2

Solution of Two Real Life Problems

To show the efficiency of algorithm SOMGA, it has been tested to solve two real life problems from the field of engineering. The problems and their solutions are described as follows.

4.1 Optimal Thermohydraulic Performance of an Artificially Roughened Air Heater

This problem is taken from Prasad and Saini [24]. In this problem the optimal thermo hydraulic performance of an artificially roughened solar air heater is considered. Optimization of the roughness and flow parameters (p/e, e/D, R_e) is considered to maximize the heat transfer while keeping the friction losses to be minimum. This is an unconstrained optimization problem. It has three decision variables. The mathematical model of the problem, as given in Prasad and Saini [24] is:

$$\text{Maximize } L = 2.5\log e^{+} + 5.5 - 0.1R_M - G_H$$

where

$$\begin{aligned}
R_M &= 0.95x_2^{0.53}\\
G_H &= 4.5(e^{+})^{0.28}(0.7)^{0.57}\\
e^{+} &= x_1x_3\left(\bar{f}/2\right)^{1/2}\\
\bar{f} &= (f_s + f_r)/2\\
f_s &= 0.079x_3^{-0.25}\\
f_r &= 2\left[0.95x_3^{0.53} + 2.5\log(1/2x_1)^2 - 3.75\right]^{-2}
\end{aligned}$$

The notations used are as follows:

e^{+}	roughness height.
p	pitch of the roughness element.
D	the hydraulic diameter of solar heater.
$x_1 = e/D$	relative roughness height.
$x_2 = p/e$	relative roughness pitch.
$x_3 = R_e$	Reynolds number.

Table 5 Optimal thermo hydraulic performance of an artificially roughened air heater

	Value of objective	Values of variables
Solution obtained by SOMGA	4.18241	$x_1 = 0.10558676$, $x_2 = 10.000276$, $x_3 = 4567.99$
Solution given in Pant [25]	4.182	$x_1 = 0.052$, $x_2 = 10.00$, $x_3 = 10258.46$
Solution given in Prasad and Saini [24]	≈4.18	$x_1 \approx 0.0205$, $x_2 \approx 10.00$

The bounds on the variables are:

$$0.02 \le x_1 \le 0.8, \quad 10 \le x_2 \le 40, \quad 3000 \le x_3 \le 20{,}000$$

This is an unconstrained problem and has been solved by SOMGA. It is earlier solved by Prasad and Saini [24] and Pant [25]. The results obtained by SOMGA are compared with the available results and are presented in Table 5. It is clear from the Table 5 that the solution obtained by SOMGA is better than previously quoted results.

4.2 *Frequency Modulation Sounds Parameter Identification Problem*

This problem has been taken from Tsutsui and Fujimoto [26]. It is an unconstrained optimization problem. This problem is to determine the six parameters $a_1,\ \omega_1,\ a_2,\ \omega_2,\ a_3,\ \omega_3$ of the Frequency Modulation Sound model represented by

$$y(t) = a_1 \cdot \sin(\omega_1 \cdot t \cdot \theta + a_2 \cdot \sin(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin(\omega_3 \cdot t \cdot \theta))),$$

with $\theta = \frac{2\cdot\pi}{100}$. An evaluation function P_{fms} is defined as the summation of 101 square errors between the evolved data and the model data as follows:

$$P_{fms}(a_1, \omega_1, a_2, \omega_2, a_3, \omega_3) = \sum_{t=0}^{100} (y(t) - y_0(t))^2,$$

where the model data are given by the following equation:

$$y_0(t) = 1.0 \cdot \sin(5.0 \cdot t \cdot \theta - 1.5 \cdot \sin(4.8 \cdot t \cdot \theta + 2.0 \cdot \sin(4.9 \cdot t \cdot \theta))).$$

Each parameter is in the range −6.40 to 6.35. In this problem, a generated sound wave and its evaluation function P_{fms} are extremely sensitive to some of these parameters a_1, w_1, a_2, w_2, a_3, w_3. This makes the problem difficult to reach the optimal point. This problem is a highly complex multimodal one having strong epistasis, with minimum value $P_{fms}(x^*) = 0$. Empirical results are shown in Table 6.

Table 6 Solution of Frequency modulation sounds parameter identification problem

Method	Result obtained	Total runs required	# Evaluation
SOMGA	0.00941009	156	7207
p-fGA	–	22621	–

Tsutsui and Fujimoto [26] have used this problem for the testing of their technique that is Phenotypic Forking Genetic Algorithm (p-fGA). This technique takes 22621 generations on average to solve this problem. On the other hand SOMGA solves this problem in 156 generation or 7207 function evaluation only.

5 Section 3

5.1 Comparison with the Memetic Algorithm

Generally two kinds of local search methods can be used in GAs to improve the efficiency of these algorithms. One is the deterministic method and the other is to use another population based stochastic search strategies as local searches. Although deterministic methods have less exploration capabilities but they converges very fast in small neighborhood. In this section a comparison has been made between the two approaches of hybridization. One approach is SOMGA described in Sect. 2, Previous work done, in which GA is hybridized with another population based technique SOMA. Another approach is hybridization of GA with a deterministic local search method. This approach is based on the idea that first let the population evolve using GA and when the search domain become narrow and convergence slow down after certain number of generations apply local search on selected individual to faster the convergence of it. The methodology of this approach is as follows:

Step 1: Generate the initial population.
Step 2: Evaluate all individuals.
Step 3: Apply selection operator on all individuals to select the better individuals for the next generation.
Step 4: Apply crossover operator on all individuals with crossover probability P_c to produce new individuals.
Step 5: Evaluate the new individuals.
Step 6: Apply mutation operator on every bit of every individual of the population with mutation probability P_m.
Step 7: Evaluate the mutated individuals.
Step 8: If generation is less than the specified generations go to Step 3 else go to Step 9.
Step 9: Select best q% individuals and apply LS.
Step 10: If termination criterion is satisfied stop else go to Step 3.

Table 7 Parameters of MA

Dimension	n = 10
Population size	100
Crossover rate p_c	0.95
Mutation rate p_m	0.001
Specified number of generations	100
Total number of function calls allowed	40,000

Table 8 Comparative results of SOMGA and MA

Problem	SOMGA				MA			
	Mean	S.D.	# of success	# of evaluation	Mean	S.D.	# of success	# of evaluation
Ackley	0.007	0.002	30	9644	0.06023	0.30033	28	22,521
Schwefel	0.004	0.003	30	11,660	0.00013	0	30	21,634
Griewank	0.068	0.036	6	13,639	0.04943	0.04123	5	33,143
Restrigin	0.003	0.003	30	16,895	0.09532	0.26137	26	32,465
Rosenbrock	4.699	1.889	0	–	0.00103	0.00167	30	21,012

The algorithm terminates as soon as the best fit solution is obtained within 1 % accuracy of the known global optimum solution or it crosses the allowed number of function evaluations. Any derivative free local search method can be used in this approach. We use the well known Hooke and Jeeves direct search method described in Bazaraa et al. [27]. It is multidimensional search method with discrete steps and works without using derivatives. Hence it is able to solve a wide range of problems. It is also very quick and robust at finding the local optimal solution of a problem. It is tested on five well-known benchmark test problems, namely, Ackley function, Schwefel's function, Griewank's function, Restrigin's function, Rosenbrock's function taken from Ali et al. [28]. These five test problems are most commonly used for evaluating the performance of evolutionary algorithms. The parameters used in this approach namely population size, probability of crossover (P_c), probability of mutation (P_m), specified number of generations (taken in our experiments after fine-tuning) after which LS has to be activated and total number of function calls allowed are given in Table 7 and the results obtained by this approach are presented in Table 8.

5.2 *Results and Discussion*

For this approach, it is suggested that for solving an n dimensional problem, population size 5 * n should be taken and LS should be applied after 10 * n generations. But minimum population size 100 is required for 10 or more than 10 dimensional problem. All the results have been taken using these specifications. Figure 3 shows the success attained by MA and SOMGA. The greater the portion covered by an algorithm in a column, the better the method is, in that problem. It is clear from the graph that the portion covered by both the algorithms GA and SOMGA is almost equal, except in one problem. The main difference between the two is that MA is providing success at 100 population size for 10 variable problem and SOMGA is providing the same success at 10 population size for 10 variable problem. Hence on the basis of population size SOMGA is better than MA.

Figure 4 shows the function evaluations taken by MA and SOMGA in successful runs. In this graph, the lesser the portion covered by an algorithm in a column, the better the method is in that problem. It is evident from the graph that the portion covered by SOMGA is much lesser than MA. This means SOMGA required less number of function evaluations than MA for obtaining the global optimal solution of unconstrained optimization problems. Hence SOMGA is more efficient than MA.

On the basis of these results, it is concluded that SOMGA is far more superior to MA in terms of function evaluations. Hence SOMGA is recommended for solving the unconstrained nonlinear optimization problems.

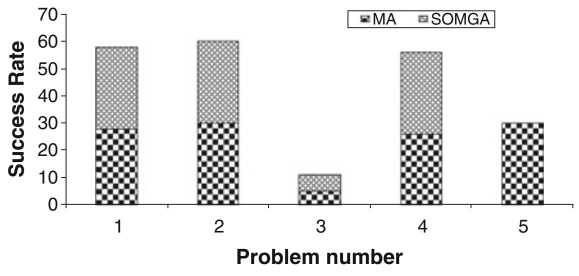

Fig. 3 Graph showing success attained by MA and SOMGA. Greater portion ⇒ better method

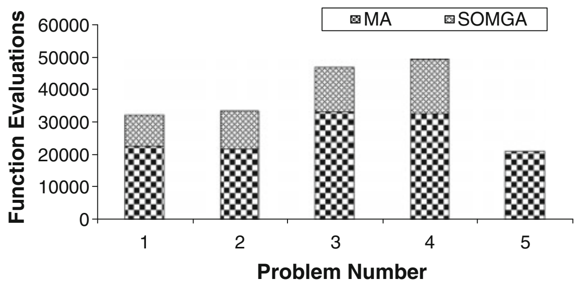

Fig. 4 Graph showing the function evaluations taken by MA and SOMGA. Lesser portion ⇒ better method

6 Conclusions

In this chapter a variant of GA and SOMA known as SOMGA has been used to solve large scale optimization problems. From the discussion part of results for large scale optimization it is clear that for solving up to 200 dimension problem SOMGA requires very less population size as well as function evaluations. Later two real life problems from the field of engineering have been solved using the same technique and SOMGA proves its efficiency to solve these problems. At last a comparison between two variants of GA i.e. SOMGA and MA has been made. Though MA is able to solve all the test problems with good success rate but it is expensive in terms of function evaluations. One drawback of using deterministic methods is that they are usually problem specific hence restricted to solve some specific classes of problems. Another drawback of using these traditional local searches is that if LS is not implemented properly, the chances of getting trapped in local minima is more because these methods have less exploration qualities. It means that these hybridized algorithms demand careful fine tuning of local search parameters. On the other hand population based stochastic search techniques are not problem specific and are applicable to solve a wide range of problems. Another advantage of using these techniques is that they use multiple guesses to improve a solution. Hence have more exploration qualities.

Appendix

This Appendix contains the list of 13 benchmark test problems taken from literature, which are used to evaluate the performance of the algorithm. These problems are unconstrained nonlinear optimization problems having a number of local as well as global optimal solutions. All the problems have varying difficulty level and contain unimodal as well as multi modal problems.

Problem 1: (Cosine Mixture Problem)
This problem is Cosine Mixture Function. The global optimum of this function is at (0, 0,…, 0) with $f_{\min} = -0.1n$. where n is the dimension of the problem. The functional form is as follows:

$$\min_x f(x) = \sum_{i=1}^{n} x_i^2 - 0.1 \sum_{i=1}^{n} \cos(5\pi x_i), \quad for\ x_i \in [-1, 1].$$

Problem 2: (Exponential Problem)
This problem is Exponential Function. The global optimum of this function is at (0, 0,…, 0) with $f_{\min} = -1$. The functional form is as follows:

$$\min_x f(x) = \exp\left(-0.5\sum_{i=1}^{n} x_i^2\right), \quad for\ x_i \in [-1,\ 1].$$

Problem 3: (Ackley Function)
This problem is the Ackley function. The surface of the Ackley function has numerous local minima due to its exponential terms. Any search algorithm based on the gradient information will be trapped in local optima, but any search strategy that analyzes a wider region will be able to cross the valley among the optima and achieve better results. Its global minimum is at (0, 0,…, 0) with $f_{\min}$ = 0. The functional form is as follows:

$$f(x) = 20 + e - 20e^{-\left(\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)} - e^{-\left(\frac{i}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)}, \quad for\ x_i \in [-15, 30].$$

Problem 4: (Sphere Function Problem)
The next problem is Sphere Function. This problem is continuous convex and unimodal. Its global minimum is at (0, 0,…, 0) with $f_{\min}$ = 0. The functional form is as follows:

$$\min_x f(x) = \sum_{i=1}^{n} x_i^2, \quad for\ x_i \in [-5.12, 5.12].$$

Problem 5: (Griewank Function)
This problem is a widely employed test function for global optimization, the Griewank function. While this function has an exponentially increasing number of local minima as its dimension increases, it turns out that a simple Multistart algorithm is able to detect its global minimum more and more easily as the dimension increases. The optima of this function are regularly distributed. Number of local minima for arbitrary n is unknown, but in two dimensional case there are some 500 local minima. Its global minimum is at (0, 0,…, 0) with $f_{\min}$ = 0. The functional form is as follows:

$$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) - 1, \quad for\ x_i \in [-5.12, 5.12].$$

Problem 6: (Axis Parallel Hyper Ellipsoid)
This problem is Axis Parallel Hyper Ellipsoid Function. This test problem is similar to sphere problem function. It is also known as the weighted sphere model. It is continuous convex and unimodal. Its global minimum is at (0, 0,…, 0) with $f_{\min}$ = 0. The functional form is as follows:

$$\min_x f(x) = \sum_{i=1}^{n} i x_i^2, \quad for\ x_i \in [-5.12, 5.12].$$

Problem 7: (Schwefel's Double Sum)
This problem is Schwefel's double sum Function. This function is an extension of axis parallel hyper ellipsoid function. It produces a rotated hype-ellipsoid. It is continuous convex and unimodal. Its global minimum is at (0, 0,…, 0) with $f_{min} = 0$. The functional form is as follows:

$$\min_x f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2, \quad for\ x_i \in [-65.536,\ 65.536].$$

Problem 8: (Rastrigin Function)
This problem is the Rastrigin Function. It is the extended form of the sphere function with a modulator term $\alpha \cdot \cos(2\pi x_i)$. This function consists of a large number of local minima (not exactly known) whose value increases with the distance to the global minimum. Its global minimum is at (0, 0,…, 0) with $f_{min} = 0$. The functional form is as follows:

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) \right), \quad for\ x_i \in [-5.12, 5.12].$$

Problem 9: (Rosenbrock Function)
This problem is the Rosenbrock function, also known as the banana function. It is a continuous, differentiable, unimodal and non separable function. Its difficulty arises due to nonlinear interaction between parameters. The global optimum is inside a long narrow parabolic shaped flat valley. Its global minimum is at (1, 1,…, 1) with $f_{min} = 0$. The functional form is as follows:

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + \left(x_i - 1\right)^2), \quad for\ x_i \in [-2.048, 2.048].$$

Problem 10: (Schwefel Function)
This problem is Schwefel Function. The contour of this function is made up of a great number of peaks and valleys. This function has a second best minimum far from the global minimum, so it is difficult for many algorithms to locate the global optimum of this function. Its global minimum is at (1, 1,…, 1) with $f_{min} = 0$. The functional form is as follows:

$$f(x) = 418.9829n - \sum_{i=1}^{n} \left(x_i \sin\sqrt{|x_i|} \right), \quad for\ x_i \in [-500, 500].$$

Problem 11: (Zakharov's Problem)

This problem is Zakharov Function. Its global minimum is at (0, 0,…, 0) with $f_{\min} = 0$. The functional form is as follows:

$$\min_x f(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} \frac{i}{2} x_i\right)^2 + \left(\sum_{i=1}^{n} \frac{i}{2} x_i\right)^4, \quad \mathit{for}\ x_i \in [-5.12, 5.12].$$

Problem 12: (Ellipsoidal Function)

This problem is Ellipsoidal Function. Its global minimum is at (1, 2,…, n) with $f_{\min} = 0$. The functional form is as follows:

$$\min_x f(x) = \sum_{i=1}^{n} (x_i - i)^2, \quad \mathit{for}\ x_i \in [-n, n].$$

Problem 13: (Schwefel Problem 4)

This problem is Schwefel Problem 4 Function. Its global minimum is at (0, 0,…, 0) with $f_{\min} = 0$. The functional form is as follows:

$$\min_x f(x) = \max_i \{|x_i|, 1 \le i \le n\}, \quad \mathit{for}\ x_i \in [-100, 100].$$

References

1. Deep, K., Singh, D.: A new hybrid self organizing migrating genetic algorithm for function optimization. In: IEEE Congress on Evolutionary Computation, pp. 2796–2803 (2007)
2. Grefensette, J.: Lamarckian learning in multi-agent environment In: Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA, Morgan Kauffman (1994)
3. Kasprzyk, G.P., Jasku, M.: Application of hybrid genetic algorithms for deconvulation of electrochemical responses in SSLSV method. J. Electroanal. Chem. **567**, 39–66 (2004)
4. Chelouah, R., Siarry, P.: A hybrid method combining continuous Tabu search and Nelder–Mead simplex algorithm for global optimization of multiminima functions. Eur. J. Oper. Res. **161**, 636–654 (2005)
5. Wang, L., Tang, F., Wu, H.: Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation. Appl. Math. Comput. **171**, 1141–1156 (2005)
6. Javadi, A., Farmani, A.R., Tan, T.P.: A hybrid intelligent genetic algorithm. Adv. Eng. Inf. **19**, 255–262 (2005)
7. Fan, S.K.S., Liang, Y.C., Zahara, E.: a genetic algorithm and a particle swarm optimizer hybridized with Nelder–Mead simplex search. Comput. Ind. Eng. **50**, 401–425 (2006)
8. Hwang, F.S., Song, H.R.: A hybrid real parameter genetic algorithm for function optimization. Adv. Eng. Inf. **20**, 7–21 (2006)
9. Zhang, G., Lu, H.: Hybrid real coded genetic algorithm with quasi-simplex technique. Int. J. Comput. Sci. Netw. Secur. **6**(10), 246–255 (2006)
10. Wei, L., Zhao, M.: A Nitche hybrid genetic algorithm for global optimization of continuous multi modal functions. Appl. Math. Comput. **160**, 649–661 (2005)
11. Premalatha, K., Nataranjan, A.M.: Hybrid PSO and GA for global optimization. Int. J. Open Probl. Comput. Math. **2** (2009)

12. Khosravi, A., Lari, A., Addeh, J.: A new hybrid of evolutionary and conventional optimization algorithm. Appl. Math. Sci. **6**, 815–825 (2012)
13. Ghatei, S., et al.: A new hybrid algorithm for optimization using PSO and GDA. J. Basic Appl. Sci. Res. **2**, 2336–2341 (2012)
14. Esmin, A., Matwin, S.: A hybrid particle swarm optimization algorithm with genetic mutation. Int. J. Innovative Comput. Inf. Control **9**, 1919–1934 (2013)
15. Zelinka I., Lampinen, J.: SOMA-self organizing migrating algorithm. In: Mendal, 6th International Conference on Soft Computing, Brno, Czech Republic, vol. 80, issue-2, p. 214 (2000)
16. Oplatkova, Z., Zelinka, I.: Investigation on Shannon-Kotelnik theorem impact on SOMA algorithm performance. In: Proceedings 19th European Conference on Modelling and Simulation Yuri Merkuryev, Richard Zobel (2005)
17. Zelinka, I.: Analytic programming by means of soma algorithm. In: Proceeding of 8th International Conference on Soft Computing Mendel '02, Brno, Czech Republic, pp. 93–101 (2002). ISBN 80-214-2135-5
18. Nolle, L., Zelinka, I.: SOMA applied to optimum work roll profile selection in the hot rolling of wide steel. In: Proceedings of the 17th European Simulation Multiconference ESM 2003, Nottingham, UK, pp. 53–58 (2003). ISBN 3-936150-25-7, 9-11
19. Nolle, L., Zelinka, I., Hopgood, A.A., Goodyear, A.: Comparision of an self organizing migration algorithm with simulated annealing and differential evolution for automated waveform tuning. Adv. Eng. Softw. **36**, 645–653 (2005)
20. Nolle, L.: SASS applied to optimum work roll profile selection in the hot rolling of wide steel. Knowl. Based Syst. **20**(2), 203–208 (2007)
21. Zelinka, I., Lampinen, J., Nolle, L.: On the theoretical proof of convergence for a class of SOMA search algorithms. In: Proceedings of the 7th International MENDEL Conference on Soft Computing, Brno, CZ, pp. 103–110, 6–8 June 2001. ISBN 80-214-1894-X
22. Zelinka, I., Oplatkova, Z., Nolle, L.: Boolean symmetry function synthesis by means of arbitrary evolutionary algorithms—comparative study. In: Proceedings of the 18th European Simulation Multiconference ESM 2004, Magdeburg, Germany, pp. 143–148, June 2004. ISBN 3-936150-35-4, 13-14
23. Onwubolu, C.G., Babu, B.V.: New Optimization Techniques in Engineering. Springer, Heidelberg (2004). ISBN 3-540-20167-X
24. Prasad, B.N., Saini, J.S.: Optimal thermo hydraulic performance of artificially roughened solar air heaters. J. Solar Energy **47**, 91–96 (1991)
25. Pant, M.: Genetic Algorithms for Global Optimization and their Applications. Ph.D. thesis, Department of Mathematics, IIT Roorkee, Formerly University of Roorkee (2003)
26. Tsutsui, S., Fujimoto, Y.: Phenotypic forking genetic algorithm (p-fGA). In: IEEE International Conference on Evolutionary Computing (ICEC '95), Vol. 2, pp. 556–572 (1995)
27. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming Theory and Algorithms. Wiley, New York (1993)
28. Ali, M.M., Khompatraporn, C., Zabinasky, Z.: A numerical evaluation of several global optimization algorithms on selected benchmark test problems. J. Global Optim. **31**, 635–672 (2005)