

A Novel Variant of Self-Organizing Migrating Algorithm for Global Optimization

Dipti Singh, Seema Agrawal and Nidhi Singh

Abstract This paper presents a novel variant SOMAQI of population based optimization technique self organizing migrating algorithm (SOMA). This variant uses the quadratic approximation or interpolation for creating a new solution vector in search space. To validate the efficiency of this algorithm it is tested on 10 benchmark test problems and the obtained results are compared with already published results using the same quadratic approximation. On the basis of comparison it is concluded that the presented algorithm shows better performance in terms of number of population size and function mean best.

Keywords Self organizing migrating algorithm • Particle swarm optimization • Quadratic interpolation • Global optimization

1 Introduction

Hybrid technique is regarded as an effective method of improving the performance of a population based technique for solving complex optimization problems. The good exploration capabilities are often used to locate some promising zones within the wide solution space, while the local optimization methods exploit the located promising zones to achieve the best solution quickly and accurately. Many

D. Singh (✉)

Department of Applied Sciences, Gautam Buddha University, Greater Noida, India
e-mail: diptima@rediffmail.com

S. Agrawal

Department of Mathematics, S.S.V.P.G. College, Hapur, India
e-mail: Seemagrwl7@gmail.com

N. Singh

School of Engineering, Gautam Buddha University, Greater Noida, India
e-mail: nidhi@gbu.ac.in

attempts have been made in literature to hybridize population based techniques with other exiting approaches such as Quadratic Approximation. A random search technique for global optimization based on the quadratic approximation was developed by Mohan and Shankar (now Deep) [1]. Deep and Das proposed four variants of quadratic approximation based hybrid genetic algorithm for function optimization and tested its performance on the set of benchmark test problems [2]. Deep and Bansal [3] presented the hybridization of PSO with quadratic approximation operator (QA) and the presented results showed the efficiency of hybrid technique [4]. A variant of Quantum behaved Particle Swarm optimization for solving global optimization problems was presented by Millie et al. To improve the performance of real coded genetic algorithm Deep and Das [5] hybridized it with quadratic approximation. Millie et al. presented a new variant of Particle Swarm optimization named QPSO for solving global optimization problems and tested it on various benchmark problems. Analysis of results showed that the use of multiparent quadratic interpolation operator outperformed basic PSO [6]. Deep and Bansal [3] developed a variant of PSO with hybridization of quadratic approximation operator for economic dispatch problems with valve-point effects.

This paper presents a variant (SOMAQI) of Self Organizing Migrating Algorithm (SOMA) which uses the quadratic approximation or interpolation for creating a new solution vector in the search space. Quadratic Approximation is an operator which produces the minima of the quadratic curve passing through the three chosen individuals and substantially improves the performance of SOMA. SOMA is an emergent search technique based on the self-organizing behavior of groups of individuals in a social environment. Like other evolutionary algorithm it also works with a population of solutions. The main feature of this algorithm which makes it distinguished as compared to other algorithms is that no new solutions are created during the search. Instead, only the positions of the solutions are changed during a generation, called a migration loop. The proposed algorithm SOMAQI creates new point in the search space using Quadratic Interpolation in such a way that diversity of the search domain can be maintained by newly generated points and can be thoroughly exploited. The direction of the search can be a little bit guided. The Competitive- Cooperative behavior can achieve the global optimal solution with a small population size in less number of function evaluations. A set of 10 well known test problem has been used to evaluate the performance of SOMAQI.

The paper is organized as follows. In [Sect. 2](#), preliminaries are presented. In [Sect. 3](#), the proposed Algorithm SOMAQI is presented. In [Sect. 4](#), the numerical results are discussed. Finally, the paper concludes with [Sect. 5](#) drawing the conclusions of the present study.

2 Preliminaries

In this section two algorithms SOMA and PSO has been described. SOMA has been used for hybridization and PSO has been used for comparison.

2.1 Self Organizing Migrating Algorithm

Self-Organizing migrating Algorithm is a population based stochastic search technique which is based on the social behavior of a group of individuals [7, 8]. At each generation the individual with highest fitness value is known as leader and the worst is known as active. Rather than competing with each other, the active individual proceeds in the direction of the leader. This algorithm moves in migration loops and in each migration loop active individual travels a certain distance towards the leader in n steps of defined length. This path is perturbed randomly by PRT parameter. PRT vector is created before an individual proceeds towards leader. This parameter has the same effect as mutation in Genetic Algorithm (GA). It is defined in the range (0, 1). The movement of an individual is given as follows:

$$x_{i,j}^{MLnew} = x_{i,j,start}^{ML} + (x_{L,j}^{ML} - x_{i,j,start}^{ML})t PRTVector_j \quad (1)$$

where $t \in (0, \text{by step to, PathLength})$ and 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.

The computational steps of SOMA are given as follows:

- Step 1: generate initial population;
- Step 2: evaluate all individuals in the population;
- Step 3: generate PRT vector for all individuals;
- Step 4: sort all of them;
- Step 5: select the best fitness individual as leader and worst as active;
- Step 6: for active individual new positions are created using Eq. (1). Then the best position is selected and replaces the active individual by the new one;
- Step 7: if termination criterion is satisfied stop else go to step 2;
- Step 8: report the best individual as the optimal solution.

```

Initialize particles, do
Calculate fitness value of each particle
If fitness value of the particle is better than pbest set current value as the new pbest
Set gbest to the best fitness value of all particles
For each particle
Calculate particle velocity according equation (2)
Update particle position according equation (3)
End for
While termination criterion is satisfied

```

Fig. 1 PSO pseudo code

2.2 Particle Swarm Optimization

Particle swarm optimization is a population based stochastic search algorithm developed by Eberhart et al. [9]. It is inspired by the social behavior of bird flocking and fish schooling. PSO consists of a swarm of particles. Particles move towards the best position in the search space with a velocity remembering each particle's best known position (pbest) and global best known position (gbest). The velocity and position of a particle is updated according to the following equations:

$$v_i^{k+1} = wv_i^k + c_1r_1(p_i - x_i^k) + c_2r_2(p_g - x_i^k) \quad (2)$$

$$x_i^{k+1} = x_i^k + v_i^k \quad (3)$$

In which: $i = 1, \dots, N$; N -the population of the group particles; k -the maximum number of iteration; r_1, r_2 -the random values between $[0,1]$, which are used to keep the diversity of the group particles; c_1, c_2 -the learning coefficients, also are called acceleration coefficients; v_i^k —the velocity of particle i in k -th iteration; x_i^k —the position of particle i in k -th iteration; p_i —the best known position of particle i ; p_g —global best known position of the group particles. The pseudo-code of PSO is given in Fig. 1.

3 Proposed SOMAQI Algorithm

In this section SOMAQI has been presented which uses the quadratic interpolation for creating the new solution member in the search space. As discussed in the introductory section in the working of SOMA, no new solutions are created during the search instead only the positions of the solutions are changed. For maintaining the diversity of the population, new points using Quadratic Interpolation are created in the search space. The methodology of this algorithm is given as follows:

First the individuals are generated randomly. At each generation the individual with highest fitness value is selected as leader and the worst one as active individual. Now the active individual moves towards leader in n steps of defined length. The movement of this individual is given in Eq. (1). Then we again select the best and worst individual from the population. A new point is created using quadratic interpolation at the end of each generation using Eq. (4). For this we choose three particles R_1 , R_2 and R_3 , where R_1 is the leader and R_2 and R_3 are randomly chosen particles from the remaining population. This new point is accepted only if it is better than active individual and is replaced with active individual.

Mathematically, the new point $x' = (x'_1, x'_2, \dots, x'_n)$ is given as

$$x' = \frac{1}{2} \frac{[(R_2^2 - R_3^2) * f(R_1) + (R_3^2 - R_1^2) * f(R_2) + (R_1^2 - R_2^2) * f(R_3)]}{[(R_2 - R_3) * f(R_1) + (R_3 - R_1) * f(R_2) + (R_1 - R_2) * f(R_3)]} \quad (4)$$

The computational steps of SOMAQI are given as follows:

- Step 1: generate initial population;
- Step 2: evaluate all individuals in the population;
- Step 3: generate PRT vector for all individuals;
- Step 4: sort all of them;
- Step 5: select the best fitness individual as leader and worst as active;
- Step 6: for active individual new positions are created using Eq. (1). Then the best position is selected and replaces the active individual by the new one;
- Step 7: create new point by QI from R_1 , R_2 and R_3 using Eq. (4);
- Step 8: if new point is better than active replace active with the new one;
- Step 9: if termination criterion is satisfied stop else go to step2;
- Step 10: report the best individual as the optimal solution.

4 Main Results

The proposed algorithm is coded in C++ and run on a Pentium IV 2.39 GHz computer. We have tested the proposed SOMAQI algorithm on the 10 unconstrained benchmark problems given below. All the problems are highly multimodal. Since SOMAQI is probabilistic technique and rely heavily on the generation of random numbers, therefore 50 trials of each are carried out, each time using a different seed for the generation of random numbers. A run is considered to be a success if the optimum solution obtained falls within 1 % accuracy of the known global optimal solution. The stopping criteria is fixed number of iterations are performed.

$$f_1(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

$$f_2(x) = \frac{1}{4000} \sum_{i=0}^{n-1} x_i^2 - \prod_{i=0}^{n-1} \cos\left(\frac{x_i}{\sqrt{i+1}}\right) + 1$$

$$f_3(x) = \sum_{i=0}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$$

$$f_4(x) = - \sum_{i=1}^n x_i \sin\left(\sqrt{|x_i|}\right)$$

$$f_5(x) = \left(\sum_{i=0}^{n-1} (i+1)x_i^4 \right) + \text{rand}[0, 1]$$

$$f_6(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i^2)}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)$$

$$f_7(x) = \max|x_i|, 0 \leq i < n$$

$$f_8(x) = \sum_{i=0}^{n-1} \left[x_i + \frac{1}{2} \right]^2$$

$$f_9(x) = \sum_{i=0}^{n-1} |x_i| + \prod_{i=0}^{n-1} |x_i|$$

$$f_{10}(x) = \sum_{i=0}^{n-1} \left(\sum_{j=0}^i x_j \right)^2$$

Initialization range and the optimum values for the functions are given in Table 1. The obtained results for functions f_1 – f_3 are given in Table 2. Results for f_4 – f_{10} are given in Table 3 along with already published results.

We have tested functions f_1 , f_2 and f_3 for three dimension sizes 10, 20 and 30. The population size taken for all dimensions is 10. The maximum number of iterations corresponding to 10, 20 and 30 dimensions is set as 1000, 1500 and 2000. We have tested functions from f_4 to f_{10} for three dimension sizes 10, 30 and 50 and the maximum number of iterations is set as 3000 for dimension 50. For the valid comparison all the parameters are taken same as in Ref. [10].

Different values of parameter PRT is taken as 0.005, 0.05, 0.1, 0.3, 0.5 and 0.9 for different functions. A total of 50 runs for each experimental setting are conducted. It is clear from the Table 2 that for function f_1 , the proposed algorithm is showing better results than BPSO and QPSO and giving comparative results than

Table 1 Initialization range and optimum values of the functions

Function	Initialization range	Optimum
f_1	[2.56, 5.12]	0
f_2	[300, 600]	0
f_3	[15,30]	0
f_4	[-500, 500]	-418.9829*n
f_7	[-1.28, 1.28]	0
f_6	[-32, 32]	0
f_7	[-100,100]	0
f_8	[-100, 100]	0
f_9	[-10,10]	0
f_{10}	[-100, 100]	0

Table 2 Comparison results of functions f_1 - f_3 (mean best)

Function	Dim	% of success	BPSO	QPSO	Q-QPSO	SOMAQI (mean best)	Standard deviation
f_1	10	100	5.5382	5.2543	$4.915e^{-19}$	0	0
	20	96	23.1544	16.2673	$7.806e^{-19}$	$2.7e^{-011}$	0
	30	96	47.4168	31.4576	$6.071e^{-19}$	$3.2264e^{-009}$	0
f_2	10	46	0.09217	0.08331	0.062657	$9.09092e^{-05}$	0.00019719
	20	54	0.03002	0.02033	0.005091	$2.25846e^{-06}$	0.00000856
	30	58	0.01811	0.01119	0.015442	$2.72336 e^{-05}$	0.00009506
f_3	10	66	94.1276	59.4764	5.544203	5.71912	0.359886
	20	72	204.336	110.664	15.53810	16.16542	0.42757
	30	72	313.734	147.609	25.68707	26.28846	0.77181

Q-QPSO for all the dimensions taken. For function f_2 the proposed algorithm is giving better results than BPSO, QPSO and Q-QPSO for all dimensions taken. For function f_3 , though the percentage of success obtained by proposed algorithm is zero but mean best value is better than BPSO and QPSO and comparative to Q-QPSO. From Table 3, we can easily say that for functions f_4 - f_{10} the proposed algorithm outperforms BPSO, QPSO and Q-QPSO for all dimensions taken. Other existing algorithms use different population sizes for different dimensions. Whereas the main advantage of using proposed algorithm is that it works with very small population size (10) for all dimensions.

5 Conclusions

This paper presents a variant SOMAQI incorporating the concept of Quadratic Interpolation. The proposed algorithm is tested on ten standard unconstrained benchmark problems and results are compared with BPSO, QPSO and Q-QPSO.

Table 3 Comparison results of functions f_4 – f_{10} (mean best)

Function	Dim	% of Success	BPSO	QPSO	Q-QPSO	SOMA/QI (mean best)	Standard Deviation
f_4	10	94	-2389.36	-3871.03	-3898.67	-4080.191	88.47912
	30	78	-6466.18	-8967.29	-9998.00	-12274.205	156.0609
	50	92	-10473.0	-13105.9	-14783.6	-20418.080	184.8745
f_5	10	78	0.50267	0.452975	0.376021	0.00033651	0.00024
	30	64	0.61722	0.501799	0.497801	0.00028473	0.00023
	50	72	0.78832	0.598823	0.537782	0.00026446	0.00022
f_6	10	100	$6.965e^{-12}$	$4.407 e^{-15}$	$1.210 e^{-15}$	$-5.8872e^{-15}$	0
	30	100	$3.618 e^{-05}$	$7.568 e^{-12}$	$5.828 e^{-15}$	$-5.8872 e^{-15}$	0
	50	100	3.43866	$1.018 e^{-06}$	$6.267 e^{-14}$	$-5.8872 e^{-15}$	0
f_7	10	96	$3.793e^{-06}$	$1.0981 e^{-09}$	$7.937 e^{-13}$	$2.7643 e^{-15}$	0
	30	94	7.836	4.11969	0.005714	$3.8742 e^{-05}$	0.000023
	50	90	27.8394	21.1619	0.2213	0.00279532	0.005437
f_8	10	100	0.00000	0.00000	0.00000	0	0
	30	100	0.05	0.00000	0.00000	0	0
	50	100	1.7	0.1	0.00000	0	0
f_9	10	100	$2.414 e^{-14}$	$2.0205 e^{-32}$	$3.327e^{-34}$	0	0
	30	100	$2.047e^{-07}$	$3.2466 e^{-13}$	$7.943 e^{-16}$	0	0
	50	100	7.73949	$1.6757 e^{-08}$	$8.667 e^{-11}$	0	0
f_{10}	10	100	$2.515 e^{-22}$	$5.5161 e^{-54}$	$1.009 e^{-62}$	0	0
	30	100	$1.343e^{-07}$	$1.4434 e^{-21}$	$2.999 e^{-27}$	$3.0790 e^{-33}$	0
	50	88	0.000176	$1.0679 e^{-10}$	$3.786 e^{-17}$	$2.9746 e^{-18}$	0

The results are obtained by using population size 10 only for all dimensions. On the basis of the results and discussion it is concluded that the proposed algorithm outperforms BPSO, QPSO and Q-QPSO in terms of mean best and population size.

References

1. Mohan, C., Shankar, K.: Random search technique for global optimization. *Asia. Pac. JOR* **11**, 93–101 (1994)
2. Deep, K., Das, K.: Quadratic approximation based hybrid genetic algorithm for function optimization: original research article. *Appl. math. Comput.* **203**, 86–98 (2008)
3. Deep, K., Bansal, J.C.: Quadratic approximation PSO for economic dispatch problems with valve-point effects. In: *International Conference on Swarm, Evolutionary and Memetic Computing*, SRM University, Chennai, 460–467 Springer (2010)
4. Deep, K., Bansal, J.C.: Hybridization of particle swarm optimization with quadratic approximation. *Opsearch.* **46**, 3–24 (2009)
5. Deep, K., Das, K.: Quadratic approximation based hybrid genetic algorithm function optimization. *Appl. Math. Comput.* **203**, 86–98 (2008)
6. Pant, M., Thangaraj, R., Abraham, A.: A new PSO algorithm with crossover operator for global optimization problems: 2nd international symposium on hybrid artificial intelligent systems (HAIS’07), softcomputing series. In: Corchado, E. et al. (ed.) *Innovations in Hybrid Intelligent Systems*, vol. 44, pp. 215–222. Springer Verlag, Germany (2007)
7. Zelinka, I., Lampinen, J.: SOMA—self organizing migrating algorithm. In: *Proceedings of the 6th International Mendel Conference on Soft Computing*, pp. 177–187 Brno (2000)
8. Zelinka, I.: SOMA—self organizing migrating algorithm. In: Onwubolu, G.C., Babu, B.V (eds.) *New Optimization Techniques in Engineering*. Springer, Berlin (2004)

9. Eberhart, R.C., Kennedy, J.: Particle swarm optimization. In: Proceedings IEEE International Conference on Neural Networks, IEEE Service Center, Piscataway, vol. 4, pp. 1942–1948, Perth (1995)
10. Pant, M., Thangaraj, R., Abraham, A.: A new quantum behaved particle swarm optimization. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO, 87–94 (2008)