Self-Organizing Migrating Algorithm for the 100-Digit Challenge

Quoc Bao Diep
VSB - Technical University of Ostrava
Ostrava, Czech Republic
diepquocbao@gmail.com

Ivan Zelinka
VSB - Technical University of Ostrava
Ostrava, Czech Republic
ivan.zelinka@vsb.cz

Swagatam Das
Indian Statistical Institute
Kolkata, India
swagatam.das@isical.ac.in

ABSTRACT

In this paper, we apply the SOMA T3A algorithm to solve 10 hard problems of the 100-Digit Challenge of the GECCO 2019 Competition. With effective improvements in choosing Migrants and Leader in the organization process, as well as the Step and PRT adaptive parameters in the migration process, the algorithm has achieved promising results. The total score that the algorithm achieved is 92.04 points.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms; Bioinspired optimization**;

KEYWORDS

Self-organizing migrating algorithm, SOMA, swarm intelligence

ACM Reference Format:
Quoc Bao Diep, Ivan Zelinka, and Swagatam Das. 2019. Self-Organizing Migrating Algorithm for the 100-Digit Challenge. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19)*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3326750

1 INTRODUCTION

The Self-organizing migrating algorithm (SOMA) was first introduced in 1999 [9]. And then, SOMA was developed by other researchers with different versions such as C-SOMGA [3], SOMAQI [8], HSOMA [6], mNM-SOMA [1], and CSOMA [5] and applied to many different problems such as obstacle avoidance for robot [2] and game [10]. Two original versions of the SOMA family are well-known and wider used are SOMA AllToOne and AllToAll (Adaptive) because of their simplicity and effectiveness.

After two decades since its appearance, problems arising in practice are increasingly complex and more diverse. Therefore the variant versions of SOMA were proposed to adapt and deal with these novel problems. One of the most recent version is the SOMA Team To Team Adaptive (SOMA T3A) [4]. In this version, the author significantly improved the performance of SOMA such as dividing the algorithm into three phases, including proposing adaptive parameters in each migration loop.

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326750

However, the 100-Digit Challenge [7] is another problem with higher complexity. It consists of 10 hard problems, and participants will achieve 10 points when completely solving each problem, which means reaching 10-digit accuracy. One of the differences from previous competitions, which comes from the requirement for accuracy in many real-world scenarios, is that there is no limit to the computational time, which means that it requires results to be more accurate than the solving time.

In this paper, we apply the SOMA T3A to solve 10 functions of the 100-Digit Challenge of the GECCO 2019 Competition.

2 THE SOMA T3A ALGORITHM

This section briefly describes the Self-organizing migrating algorithm Team To Team Adaptive (SOMA T3A) [4]. At the beginning, a population is randomly generated, as described in Eq. 1, and then they are evaluated by the given fitness function.

$$Pop = x_j^{(lo)} + rand[0,1](x_j^{(hi)} - x_j^{(lo)}) \tag{1}$$

where:

- Pop: the initial population of the algorithm,
- $x_j^{(lo)}, x_j^{(hi)}$: the lowest and highest boundary values,
- $rand[0,1]$: random number from 0 to 1.

After that, the algorithm randomly selects a small group containing m individuals in the population, and then selects the best n out of m individuals to become Migrants. To select the Leader, algorithm randomly select a small group containing k individuals in the population, and then select the best individual from k, this individual is the Leader. Migrants will jump step-by-step towards the Leader. Their jumping process is described in Eq. 2.

$$x_{n,j}^{ML+1} = x_{c,j}^{ML} + (x_{l,j}^{ML} - x_{c,j}^{ML})\ t\ PRTVector_j \tag{2}$$

where:

- $x_{n,j}^{ML+1}$: the new position in the next migration loop,
- $x_{c,j}^{ML}$: the position in current migration loop,
- $x_{l,j}^{ML}$: the leader position in current migration loop,
- t: jumping step.

$$Step = 0.02 + 0.005\ cos(0.5\pi\ 10^{-7}\ FEs) \tag{3}$$

$$PRT = 0.33 + 0.25\ cos(2\pi\ T\ 10^{-7}\ FEs + \pi) \tag{4}$$

The $Step$ specifies the granularity of the migration process and given in Eq. 3. The PRT causes the individual instead of moving straight to the Leader, it will move in an $N - k$ dimensional subspace and given in Eq. 4. Before each jumping, a random number is generated for each dimension and compared to PRT. If it is smaller than PRT, $PRTVector_j$ is set to 1 otherwise set to 0 (see Eq. 5).

Table 1: The tuned parameters.

Function	1	2	3	4	5	6	7	8	9	10
PopSize	150	150	150	150	150	150	150	1500	150	150
T	1	0.02	1	1	1	1	0.1	0.012	0.3	1

Table 2: Fifty runs for each function sorted by the number of correct digits

F	Number of correct digits											Score
	0	1	2	3	4	5	6	7	8	9	10	
1	50	50	50	50	50	50	50	50	50	50	50	10
2	50	50	50	50	50	50	50	50	50	50	50	10
3	50	50	50	50	50	50	50	50	50	50	50	10
4	50	50	50	50	50	50	50	50	50	50	50	10
5	50	50	50	50	50	50	50	50	50	50	50	10
6	50	50	50	50	50	50	50	50	50	50	50	10
7	50	50	50	48	48	48	48	48	48	48	48	10
8	50	50	50	22	22	22	22	22	22	22	22	9.04
9	50	50	50	41	0	0	0	0	0	0	0	3
10	50	50	50	50	50	50	50	50	50	50	50	10
											Total:	92.04

$$if\ rand_j < PRT;\ PRTVector_j = 1;\ \ else,\ 0. \tag{5}$$

After each individual finishes its movement, if the best position on its path is better than the original position, it will replace the original position. Otherwise, it will be ignored. A new migration loop is then started and run until reaching the given stop conditions.

3 EXPERIMENT SETUP

Implementation environment: The SOMA T3A was coded on Matlab and was implemented on 10 fitness functions from the 100-Digit Challenge of the GECCO 2019 competition [1] [7] . All test functions are scalable and have the same global minimum value within the given search range ($F_i^* = 1.000000000$).

Scores: For each function, the algorithm will be run 50 consecutive trials, and the total number of correct digits in the 25 trials that have the best values will be counted. The score, for each function, is the average number of correct digits in that 25 best trials [7]. For example, the algorithm reaches more than 50% of the trials achieving all 10 digits, then the score of that function is 10 points. The highest score is 100 points.

Termination criterion: The algorithm will terminate when one of the two following criteria is met:

- Reaching the 10-digit level accuracy,
- Reaching the $MaxFEs$ ($MaxFEs = 4 \times 10^8$ for all functions).

SOMA T3A parameters: The fixed parameters that were held constant value for all 10 functions are the $Step$ (given in Eq. 3), the number of jumps ($N_{jumps} = 100$), and the number of individuals of the group m ($m = 5$), n ($n = 4$), and k ($k = 5$).

Two tuned parameters are the $PopSize$ and T value in the PRT (given in Eq. 4). Their values, corresponding to each problem, are given in Table 1.

[1]The source code is publicly available at https://www.mathworks.com/matlabcentral/fileexchange/71328-soma-t3a-for-the-100-digit-challenge-gecco-2019.

4 SIMULATION RESULTS

Table 2 shows the calculation results. The first column lists the sequence number of functions, the Number of correct digits columns count the number of trials in a run of 50 times that the algorithm achieved n correct digits (with n = 1 to 10), corresponding to each function. The final column shows the average number of correct digits of the best 25 runs, which is also the score of that function. The last row is the total score for 10 functions.

In this table, for functions 1, 2, 3, 4, 5, 6, and 10, the algorithm reached 50 over 50 runs achieving 10 correct digits; the score for each of them is 10 points. For functions 7, the algorithm has 48 times achieving 10 correct digits; the score for function 7 is 10 points.

For functions 8, the algorithm has 50 times achieving 2 correct digits and 22 times achieving 10 correct digits; the score for this function is 9.04 points.

For function 9, the algorithm has 50 times achieving 2 correct digits and 41 times achieving 3 correct digits, there are no times in 50 runs that the algorithm achieves above 3 correct digits; the score for this function is 3 points.

Therefore, the total score is 92.04 points.

5 CONCLUSIONS

The SOMA T3A algorithm has achieved promising results, with 92.04 points in total. For functions 1 to 7 and function 10, the algorithm has achieved 10-digit accuracy over 25 trial runs (i.e. 10 points for each function). Functions 8 and 9 are hard problems for SOMA T3A when it only reached 9.04 and 3 points respectively.

ACKNOWLEDGMENTS

The following grants are acknowledged for the financial support provided for this research: SGS SP2019/137, VSB-Technical University of Ostrava.

REFERENCES

[1] Seema Agrawal and Dipti Singh. 2017. Modified Nelder-Mead self organizing migrating algorithm for function optimization and its application. *Applied Soft Computing* 51 (2017), 341–350.
[2] Diep Quoc Bao and Ivan Zelinka. 2019. Obstacle Avoidance for Swarm Robot Based on Self-Organizing Migrating Algorithm. *Procedia Computer Science* 150 (2019), 425–432.
[3] Kusum Deep and Dipti. 2008. A self-organizing migrating genetic algorithm for constrained optimization. *Appl. Math. Comput.* 198, 1 (2008), 237–250.
[4] Quoc Bao Diep. [n. d.]. Self-Organizing Migrating Algorithm Team To Team Adaptive – SOMA T3A. In *The 2019 IEEE Congress on Evolutionary Computation, Wellington, New Zealand.* accpeted, in print.
[5] Leandro dos Santos Coelho and Viviana Cocco Mariani. 2010. An efficient cultural self-organizing migrating strategy for economic dispatch optimization with valve-point effect. *Energy Conversion and Management* 51, 12 (2010), 2580–2587.
[6] Zhiyi Lin and Li juan Wang. 2014. Hybrid self-organizing migrating algorithm based on estimation of distribution. In *2014 International Conference on Mechatronics, Electronic, Industrial and Control Engineering (MEIC-14).* Atlantis Press.
[7] K. V. Price, N. H. Awad, M. Z. Ali, and P. N. Suganthan. November, 2018. Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization. In *Technical Report, Nanyang Technological University, Singapore.*
[8] Dipti Singh and Seema Agrawal. 2016. Self organizing migrating algorithm with quadratic interpolation for solving large scale global optimization problems. *Applied Soft Computing* 38 (2016), 1040–1048.
[9] Ivan Zelinka and Lampinen Jouni. 2000. SOMA–Self-Organizing Migrating Algorithm Mendel. In *6th International Conference on Soft Computing, Brno, Czech Republic.*
[10] Ivan Zelinka and Lubomir Sikora. 2015. StarCraft: Brood War—Strategy powered by the SOMA swarm algorithm. In *Computational Intelligence and Games (CIG), 2015 IEEE Conference on.* IEEE, 511–516.