A novel numerical function optimization framework: campaign based optimization framework

A novel numerical function optimization framework: campaign based optimization framework

Yuntao Liang[1*], Jiangwen Xiao[1], Zhengyi Huang[1] and Tiqian Liu[1]

[1]School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China

*Corresponding author's e-mail: hustlyt@163.com

Abstract. This paper proposes a new optimization framework: campaign optimization framework (COF), which is based on the historical results of traditional swarm intelligence algorithms and uses two different campaign rules to get better solutions. Firstly, the candidate solution set is generated by running the swarm optimization algorithm. Secondly, cross campaign rule (CCR) or increment campaign rule (ICR) is used to update the candidate solution set. The campaign rule describes the steps to get better solutions. In CCR, the better solution is obtained by combining the variable bits of two randomly selected solutions. But in ICR, it is generated by every variable bits' optimal pools. Then, the best candidate solution is optimized by swarm optimization algorithm again. Finally, the proposed framework is tested on two well-known benchmark functions. Through the analysis of the experimental results, the proposed algorithm is compared with traditional swarm optimization algorithm. And two campaign rules are compared in terms of results and optimization time dimensions.

1. Introduction

In recent decades, researches on swarm intelligence have never stopped. All of swarm intelligence algorithms can be broadly divided into two categories: one is Evolutionary Algorithm(EA), such as Genetic Algorithm(GA)[1-2], Artificial Immune Algorithm(AIA)[3] and Quantum genetic algorithm (QGA)[4], the other is Swarm Intelligence(SI), such as Artificial Bee Colony Algorithm(ABC)[5], Particle Swarm Optimization(PSO)[6-7], Artificial Fish Swarm Algorithm(AFSA)[8] and Ant Colony Optimization(ACO)[9]. The main reason for the popularity of swarm intelligence optimization is its good off-the-shelf performance. However, traditional swarm intelligence optimization algorithms only give a feasible near-optimal solution[10]. The results of each run of them are not so stable because the traditional algorithms run independently each time. Usually, we run traditional algorithms many times and merely select the best one among them.

In this paper, we proposed a novel swarm optimization framework: Campaign based Optimization Framework (COF). This model is mainly inspired by the phenomenon of political campaigns in human society and fundamentally different from the previous swarm intelligence algorithms. In a political campaign, a candidate elected by voters only represents these voters' interests, and can get higher support by adjusting their political views in a timely manner through various political tests. After several rounds of elections, the final winner is chosen as an optimal solution.

We attempt to simulate the phenomenon of political campaigns in human society to solve the optimization problem. First, we abstract the solution vector into the policy propositions given by the candidates. Different variables correspond to different problems, and the values of the variables are equivalent to the solutions given by the candidates. If the claims put forward by the candidate meet most

people's requirements, then the candidate will have higher support, in which the support degree corresponds to the objective function value in the optimization problem. We regard the traditional swarm intelligence optimization process as the module of voter election, and each time the best solution is equivalent to a candidate (also called candidate solution later). Running this algorithm multiple times will yield a candidate solution set. Then, we should institute rules for module of campaign. In this module, each candidate can modify his own policy according to the policy of other candidates, so that he achieves a higher degree of support. The campaign can be conducted several rounds, and each round is going to choose one who has a higher degree of support. In the last round of election, the final winner is chosen, that is the ultimate solution. In fact, the campaign framework can be interpreted as that converting the general continuous optimization problem into a combinatorial optimization problem through the module of campaign. The election process is a combination of candidate solutions with higher support. We refer to these approaches as COF.

There are several salient advantages of COF. 1) COF can combine the results of traditional algorithms to get better solutions. 2) COF can be implemented by parallel encoding easily, which can improve efficiency and save time as a matter of course. 3) COF can flexibly be framed based on practical problems. This paper makes the following contributions:

Through political campaigns to explain the COF proposed in this paper. COF is easy to employ an existing swarm optimization algorithm and to obtain a better result.

We detail the structure of COF. COF mainly consists of parent algorithm and campaign rule. In campaign rule part, we proposed two different CR in module of campaign of COF.

We report experimental results to evaluate the proposed COF. Two classical test functions are used to detect the optimal performance of COF.

2. Proposed Algorithm

The detailed steps of the proposed campaign based optimization framework(COF) are as follows:

Algorithm 1 campaign based optimization framework(COF)

```
Set parameters: Parent optimization algorithm(POA), parameters of POA,
Campaign Rule(CR), the number of campaign(NUM_CAM), the number
of candidates(NUM_CANDI)
Run POA NUM_CANDI times to get candidates set
for i=1 to NUM_CAM
    update candidates with CR
    optimize candidates with optimization algorithm
end for
Output candidates
```

2.1Parent Optimization Algorithm

The selection of the parent algorithm plays a decisive role in the performance of the whole optimization framework. The parent optimization algorithm (POA) is equivalent to general election in the political campaign. It provides candidates for the later campaign. in the other word, POA generates the candidate solution pool. The parent algorithm can be any kind of swarm intelligence optimization algorithm, such as: genetic algorithm (GA), particle swarm optimization algorithm (PSO), bee colony algorithm (ABC), fish swarm algorithm (AFSA). After the POA is determined, the following is running repeatedly POA to optimize the same problem. Because of the need to iterate the POA, and there is no iteration dependent on the results of other iterations in the loop, at this stage, parallel coding can be used to reduce runtime consumption.

2.2Campaign Rule (CR)

In this section, we introduce two novel campaign rules in the module of campaign. The two algorithms follow different approaches to achieve this goal. Most candidate solution vectors have better values than

other candidate solutions on some same positions, and the optimal variable positions of different candidate solutions may be different. The objective of the campaign is to find the optimal solution of all candidate solution for each variable positions, which can be combined to a better candidate solution.

2.2.1Cross Campaign Rule (CCR)

First, two candidate solutions are selected randomly, and then the solutions corresponding to each variable bit are determined randomly according to the two candidate solutions. Then we compute the support of the new composite solution. If the support of the current candidate solution is greater than the minimum support of the previous candidate solution, the candidate solution with the minimum support is replaced by the solution, and an optimal solution is obtained by iteration with this method. The cross campaign rule is shown as Algorithm 2.

Algorithm 2 Cross Campaign Rule(CCR)

```
Set the number of iterate(NUM_ITER), initialize MIN_SUPPORT =
min{CANDIDATE_SUPPORTS}
for i=1 to NUM_ITER
   Randomly select two candidates Candidate_1, Candidate_2
   Generate CANDIDATE_NEW by combining with two candidates
   compute NEW_SUPPORT of CANDIDATE_NEW
   if NEW_SUPPORT>MIN_SUPPORT then
      replace worst candidate with CANDIDATE_NEW
      replace worst support with NEW_SUPPORT
      update MIN_SUPPORT
   end if
end for
Output candidates
```

2.2.2Increment Campaign Rule(ICR)

In CCR, the worst candidate solution is replaced by the slightly better solution that combines with two existent candidates, and the number of candidate solutions is not changed. The distinguished point of ICR is to continuously add the best solution to the optimization pool, and naturally to continuously improve the probability of being selected by some mechanisms. The ICR is not the same as that of the CCR, which is a completely random selection of candidate solutions to generate new solutions.

The detail process of ICR is summarized as follows. Input the candidate solution obtained by general election and remove the original combination relation of the candidate solutions. Then the values of the same variable bits of all candidate solutions are put into an optimal pool, and there are optimal solutions, where n is the number of dimensions of the problem. Then a clustering algorithm is used to group each optimal pool, and the variables in the optimal pool are subdivided into several sub optimal pools. Each large pool is composed of a number of sub optimal pools. Then, probability tables of each optimization pool are initialized with same probability. The probability table will be adjusted if the support of the new candidate solution is better than worst one. Choose the corresponding sub optimal pool according to probability table, and then randomly selects a values from the sub optimal pool to each variable bit, finally, we use them together to form a new candidate solution. Calculate the support of new candidate solutions support, if it is less than the current optimal support, we start a new round of campaign, or subsequent corresponding operation, an operation is the probability table adjustment: the probability of sub optimal solutions corresponding to each optimal solution pool increase in the pool. Another one is to copy values of each variable bit of new candidate over corresponding sub optimal pool, this operation is to improve the probability of optimal value in sub optimal pool. After these two operations, continue a new round of campaign. When the election meets the termination condition, stop campaigning and export the campaign results. The ICR is shown as Algorithm 3.

Algorithm 3 Increment Campaign Rule(ICR)

```
Set the number of iterate(NUM_ITER), initialize MAX_SUPPORT =
max{CANDIDATE_SUPPORTS}
Use clustering algorithm to group each optimal pool and initialized the
probability table(PROB_T).
for i=1 to NUM_ITER
   randomly select the corresponding sub optimal pool according to PROB_T
   randomly select a value from the sub optimal pool to each variable bit
   Generate CANDIDATE_NEW by combining with each variable bit
   according to PROB_T
   compute NEW_SUPPORT of CANDIDATE_NEW
   if NEW_SUPPORT>MAX_SUPPORT then
      update MAX_SUPPORT, PROB_T
      add each variable bit of CANDIDATE_NEW to sub optimal pool
   end if
end for
Output candidates
```

2.3Optimizing candidates

After the module of campaign, we have completed the optimization of the results of the general election, and assembled more excellent candidate solutions. In fact, this new candidate solution can be further optimized by some special algorithms. There are many ways to optimize the candidate solution, which can be some algorithm with local search ability strong, such as taboo search, swarm intelligence algorithm, POA also.

3. Experiment results

In this section, we report experimental results to evaluate the proposed COF algorithm. For comparison, we also evaluate tree state-of-the-art swarm intelligence algorithm including GA, PSO and ABC, which have very high optimization performance.

3.1Benchmark function

In order to compare the performance of the proposed ICR and CCR with its parent optimization algorithm (POA), we used two classical benchmark functions as given in [5].

The first function is Griewank function whose value is 0 at its global minimum (0, 0, …, 0). Initialization range for the function is [−600,600]. The second function is Rastrigin function whose value is 0 at its global minimum (0, 0, …, 0). Initialization range for the function is [−15,15].

$$\begin{aligned} fun_1\left(\vec{x}\right) &= \frac{1}{400}\left(\sum_{i=1}^{D}(x_i^2)\right)-\left(\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)+1 \\ fun_2\left(\vec{x}\right) &= \sum_{i=1}^{D}\left(x_i^2-10\cos\left(2\pi x_i\right)+10\right) \end{aligned} \tag{1}$$

3.2Parameters setting

We can select GA, PSO or ABC as POA, the parameters for POA are listed as follows:

- GA: the number of generation is 1000, the number of population is 100, the intersection probability is 0.3, the variation probability is 0.1.
- PSO: the number of generation is 100, the number of population is 100, the inertia weight is random value, the speed regulation is [1.4944,1.4944].
- ABC: the number of generation is 1000, the number of population is 100, the fluorescein volatilization factor is 0.4, the speed regulation is [1.4944,1.4944].

Results with dimensions of 5, 10, 20, 40 are recorded separately. The number of iteration is variable according to the dimension of problem. The number of campaign is 2. That is, the number of candidates change from 30 to 8, and only one success.

3.3Results

We select PSO as POA, then generate 30 candidate solutions by PSO, set the dimension be 5.

3.3.1CCR

Table 1 shows the value of candidate solutions before and after the campaign. Through the table, we find that after the COF, the value of each candidate solution tends to be consistent. This reduces the richness of the solution.

Table 1. Optimization results of CCR before and after campaign

	Variable bit	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6	Solution 7	Solution 8	Solution 9	Solution 10
Before campaign	X_1	-12.5604	-3.1398	0.0000	-6.2796	-6.2782	-6.2800	-12.5599	3.1407	-0.0002	-12.5608
	X_2	8.8760	-4.4383	-13.3153	-4.4383	-13.3159	-4.4387	-13.3152	4.4374	0.0005	-0.0003
	X_3	0.0002	-5.4325	-5.4320	-0.0009	10.8671	0.0001	5.4332	-5.4330	-10.8665	-5.4330
	X_4	-12.5413	-6.2705	-6.2720	-12.5410	6.2697	-6.2706	-12.5418	-12.5406	-6.2709	6.2706
	X_5	-14.0152	-0.0007	-7.0068	-7.0032	-0.0050	-0.0007	0.0009	-7.0074	-7.0058	-0.0012
	F(X)	-0.1478	-0.0246	-0.0739	-0.0665	-0.0937	-0.0246	-0.1306	-0.0665	-0.0517	-0.0567
After campaign	X_1	-3.1398	-0.0002	-0.0002	-0.0002	-0.0002	-3.1398	-0.0002	-3.1398	-3.1398	-0.0002
	X_2	-4.4383	0.0005	0.0005	0.0005	0.0005	-4.4383	0.0005	-4.4383	-4.4383	0.0005
	X_3	-5.4325	-5.4325	-5.4325	-5.4325	-5.4325	-5.4325	-5.4325	-5.4325	-5.4325	-5.4330
	X_4	6.2706	-6.2705	-6.2705	-6.2705	-6.2705	6.2706	-6.2705	-6.2705	-6.2705	6.2706
	X_5	-0.0007	-0.0007	-0.0007	-0.0007	-0.0007	-0.0007	-0.0007	-0.0007	-0.0007	-0.0012
	F(X)	-0.0246	-0.0172	-0.0172	-0.0172	-0.0172	-0.0246	-0.0172	-0.0246	-0.0246	-0.0172

3.3.2ICR

In ICR, the value of probability increment can be a fixed value, and can also be a variable. We only discuss the probability increase of fixed value of the case in this paper. If the value of probability increment is large, this will increase the speed of convergence. For early iteration searching speed is faster, but late in the iteration, it takes a long time to jump out of local optimum. If the value of increasing probability is small, this will Slow the speed of convergence. Through the experiment, when the value of probability increment equal to the original probability, the searching ability of COF is better. The result of experiments is given in Figure 1 and Figure 2.

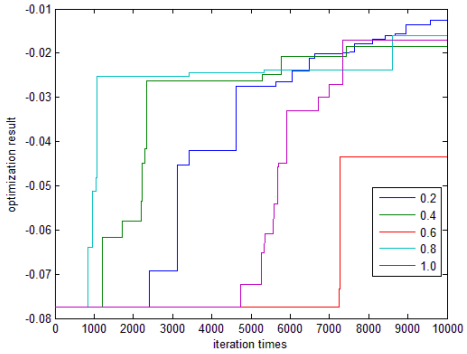

Figure 1. Optimization results for Rastrigin function of the first 10,000 iterations under different probability increments

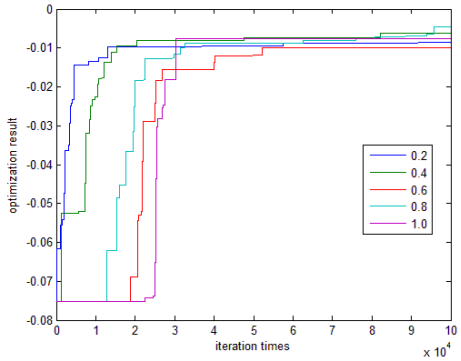

Figure 2. Optimization results for Rastrigin function of the first 100,000 iterations under different probability increments

3.3.3Comparison between two rules

Due to the different iterative mechanism, the performance of the two campaign rules is also different. Specifically, the convergence rate of CCR is fast, but it is easy to fall into local optimization. ICR can get a better solution, but it needs to iterate many times. Especially when the dimension of the problem is very large, it is difficult to effectively optimize the results. In general, the CCR can be used to build a campaign framework to achieve the optimization of the solution, while reducing the time consumption.

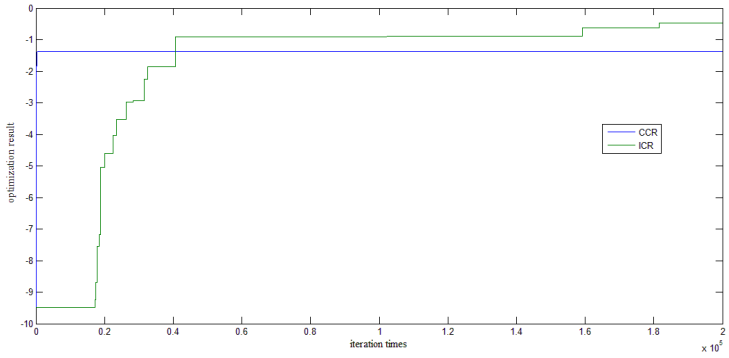

Figure 3. Optimization results for Rastrigin function of the first 100,000 iterations under different campaign rules with 20 dimension.

3.3.4Details

Details are listed in appendices for every POA, campaign rule, benchmark function. It can be seen from the results that both campaign rules can get better solutions. Because CCR rules are more random and reduce the diversity of understanding, ICR is better when the dimension is higher.

4. Conclusions

This paper has introduced a campaign optimization framework (COF) inspired by political campaign in human society. In COF, POA and the optimization algorithm after updating candidates with CR are variable, which increases the flexibility of the framework. Based on the results of traditional swarm optimization algorithms, CCR and ICR improve the capability of escaping local optimum by the combination and exchange of candidate solutions. From experimental results of two classical benchmark functions, the following conclusions can be drawn. First, COF is feasible and effective for different POAs and dimensions. Moreover, as the core of COF, both CCR and ICR have their own advantages. ICR works better when the dimension is higher, and CCR is more effective in terms of convergence speed.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 61773172, 61572210, and 51537003, the Natural Science Foundation of Hubei Province of China (2017CFA035), the Fundamental Research Funds for the Central Universities (2018KFYYXJJ119) and the Program for HUST Academic Frontier Youth Team.

Appendices

In table 2 and table 3, GA_ICR represents that the POA is GA and the campaign rule is ICR, Dim is the dimension of the problem, POA_best is the result of POA, result_1 is the result after the first campaign, result_2 is the final result.

Table 2. Optimization results for Griewank function

Algorithm	Dim	POA_best	Result_1	Result_2	Algorithm	Dim	POA_best	Result_1	Result_2
GA_ICR	5	-2.95E-09	-1.35E-12	-4.07E-14	**GA_CCR**	5	-2.95E-09	-5.38E-13	-1.33E-15
GA_ICR	10	-0.0099	-5.43E-09	-1.69E-10	**GA_CCR**	10	-0.0099	-5.07E-09	-4.77E-11
GA_ICR	20	-0.015635	-6.78219E-06	-4.84094E-07	**GA_CCR**	20	-0.015635	-2.03184E-05	-1.54495E-06
GA_ICR	40	-1.137091	-0.050722943	-0.002950193	**GA_CCR**	40	-1.137091	-0.092630183	-0.004756254
PSO_ICR	5	-4.047103	-0.08855738	-0.039237722	**PSO_CCR**	5	-4.047103	-0.090110271	-0.029889593

PSO_ICR	10	-12.89499	-1.387932524	-0.739960573	**PSO_CCR**	10	-12.89499	-0.440601937	-0.201990962
PSO_ICR	20	-32.21299	-10.19075344	-2.187362519	**PSO_CCR**	20	-32.21299	-2.1736132	-1.257494871
PSO_ICR	40	-91.06136	-49.99869193	-30.26664211	**PSO_CCR**	40	-91.06136	-19.74013853	-6.128096962
ABC_ICR	5	-0.002649	-4.02379E-06	-4.02379E-06	**ABC_CCR**	5	-0.002649	-4.02379E-06	0
ABC_ICR	10	-3.8167E	-7.7532E-07	-5.36908E-07	**ABC_CCR**	10	-3.8167E	-2.87493E-06	-6.42272E-07
ABC_ICR	20	-1.86888E	-2.67527E-09	-5.72246E-10	**ABC_CCR**	20	-1.86888E	-1.17783E-07	-1.34529E-09
ABC_ICR	40	-0.000179	-1.69994E-05	-4.40208E-06	**ABC_CCR**	40	-0.000179	-0.000179588	-6.21132E-08

Table 3. Optimization results for Rastrigin function

Algorithm	**Dim**	**POA_best**	**Result_1**	**Result_2**	**Algorithm**	**Dim**	**POA_best**	**Result_1**	**Result_2**
GA_ICR	5	-1.3997E-12	-4.2632E-14	-3.5527E-14	**GA_CCR**	5	-1.3997E-12	-9.9476E-14	-7.1054E-15
GA_ICR	10	-4.9969E-07	-2.1666E-09	-6.4233E-12	**GA_CCR**	10	-4.9969E-07	-4.3732E-10	-3.6806E-12
GA_ICR	20	-0.00617885	-5.4427E-06	-9.3258E-07	**GA_CCR**	20	-0.00617885	-2.9215E-06	-1.0686E-06
GA_ICR	40	-36.4612158	-1.41172676	-0.00868847	**GA_CCR**	40	-36.4612158	-2.38760776	-0.01305963
PSO_ICR	5	-0.00222761	0	0	**PSO_CCR**	5	-0.00222761	0	0
PSO_ICR	10	-5.99387432	-0.00078433	-0.00010070	**PSO_CCR**	10	-5.99387432	-0.00024337	-1.0060E-05
PSO_ICR	20	-22.1136782	-4.08748106	-0.02153200	**PSO_CCR**	20	-22.1136782	-0.07449677	-0.02910755
PSO_ICR	40	-100.023351	-20.7144238	-11.807266	**PSO_CCR**	40	-100.023351	-13.8921044	-12.9537395
ABC_ICR	5	0	0	0	**ABC_CCR**	5	0	0	0
ABC_ICR	10	0	0	0	**ABC_CCR**	10	0	0	0
ABC_ICR	20	-0.00012707	-3.6306E-06	-8.8238E-07	**ABC_CCR**	20	-0.00012707	-2.5601E-05	-1.7739E-06
ABC_ICR	40	-20.6116155	-6.16117757	-1.77200672	**ABC_CCR**	40	-20.6116155	-16.6809944	-0.76078312

References

[1] Maeda Y, Li Q. (2005) Parallel genetic algorithm with adaptive genetic parameters tuned by fuzzy reasoning. International Journal of Innovative Computing, Information and Control,1:95-107.

[2] Pan J S, McInnes F R, Jack M A. Application of parallel genetic algorithm and property of multiple global optima to VQ codevector index assignment for noisy channels. Electronics Letters, 1996, 32(4): 296-297.

[3] De França F O, Von Zuben F J.,2009. A dynamic artificial immune algorithm applied to challenging benchmarking problems.2009 IEEE Congress on Evolutionary Computation. Trondheim. pp.423-430.

[4] Wang L, Tang F, Wu H. (2005) Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation. Applied Mathematics and Computation, 171: 1141-1156.

[5] Karaboga D, Basturk B. (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39: 459-471.

[6] Chang J F, Roddick J F, Pan J S, et al. (2005) A parallel particle swarm optimization algorithm with communication strategies, 21:809-818.

[7] Marinakis Y, Migdalas A, Sifaleras A. (2017) A hybrid particle swarm optimization–variable neighborhood search algorithm for constrained shortest path problems. European Journal of Operational Research, 261: 819-834.

[8] Li X L, Qian J X. (2003) Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques. Journal of circuits and systems, 1: 1-6.

[9] Chen Z, Zhou S, Luo J. (2017) A robust ant colony optimization for continuous functions[J]. Expert Systems with Applications, 81: 309-320.

[10] Tomassetti G, Cagnina L. (2013) Particle swarm algorithms to solve engineering problems: a comparison of performance. Journal of Engineering,2013.