A Swarm Intelligence Algorithm Inspired by Twitter

Zhihui Lv, Furao Shen(✉), Jinxi Zhao, and Tao Zhu

National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing, China
frshen@nju.edu.cn

Abstract. For many years, evolutionary computation researchers have been trying to extract the swarm intelligence from biological systems in nature. Series of algorithms proposed by imitating animals' behaviours have established themselves as effective means for solving optimization problems. However these bio-inspired methods are not yet satisfactory enough because the behaviour models they reference, such as the foraging birds and bees, are too simple to handle different problems. In this paper, by studying a more complicated behaviour model, human's social behaviour pattern on Twitter which is an influential social media and popular among billions of users, we propose a new algorithm named Twitter Optimization (TO). TO is able to solve most of the real-parameter optimization problems by imitating human's social actions on Twitter: following, tweeting and retweeting. The experiments show that, TO has a good performance on the benchmark functions.

Keywords: Swarm Intelligence · Social Media · Twitter Optimization · Particle Swarm Optimization

1 Introduction

In recent years, bio-inspired algorithms (BIAs) have been proved to be effective for solving optimization problems by mimicking the biological behaviors and nature phenomenon. With the characteristics of low computing cost, excellent efficiency and superb performance, these algorithms [1–3] became prevalent among scientists and engineers. However, there are still some problems such as premature and curse of dimensionality to be solved in this field.

As the most intelligent bio-system, human society is considered to be very worthy for investigation. By watching human's daily routine, we surprisingly found that, we ourselves, as individuals of human society, were actually performing some optimization tasks unconsciously on Internet, or more specifically, on Twitter. This is because when we use Twitter, we tend to post the Tweet (a Twitter term similar to message) that is considered to be more valuable. For example, if someone receives two Tweets, separately about local weather and presidential election, he would be more willing to share the latter message on

A. Hirose et al. (Eds.): ICONIP 2016, Part III, LNCS 9949, pp. 344–351, 2016.
DOI: 10.1007/978-3-319-46675-0_38

Twitter because it is noteworthy. Similar filtering behaviour happens on every user of Twitter. Everyone receives the Tweets filtered by others and posts the Tweets filtered by himself. Finally the Tweets which can widely spread among the users, have been filtered millions of times and would be considered as the most valuable Tweets at that moment. This explains why Twitter users can always catch up with the hotspot. Inspired by this phenomenon, we proposed a new algorithm named Twitter Optimization (TO).

To create an effective algorithm by the phenomenon mentioned above, TO introduces three metaphors about Twitter. Firstly, every Tweet is considered as a solution vector x composed of D real-valued parameters. And for a minimize optimization mission of objective function F, the smaller the value $F(x)$ is, the more valuable the Tweet is. Secondly, when a person posts a Tweet, he will push the content to all his followers (a Twitter term represents a unidirectional relationship). This means he produces a solution and shares it to the specific crowd connected to him. Finally, each person needs to retweet (a Twitter term means reposting) the most valuable Tweet from the people he followed, which implies he is helping the better solution to spread further. The realization of the three metaphors makes TO tend to find the global optimum.

2 Twitter Optimization

As a Twitter mimicking algorithm, Algorithm 1 shows the TO framework described with Twitter terms, such as Tweet and Retweet. The realization of each term will be explained later. Besides, the goal of TO is to find the solution to minimize the target function, and it is transformed into finding the most valuable Tweet on Twitter.

2.1 Term Explanation

To make a further interpretation about the details of TO, we will explain each operations by listing the equations and parameters. The significance of every equation will be illustrated.

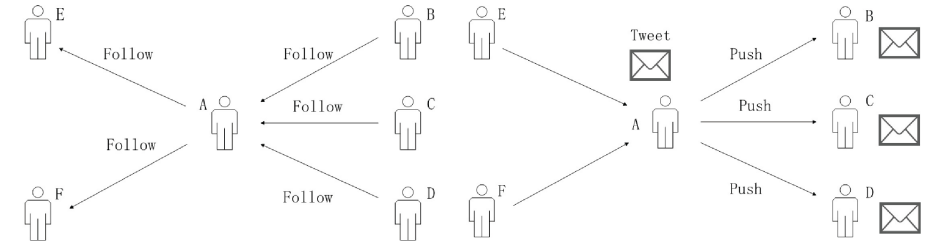
Tweet. Every Tweet is treated as a solution vector of the objective function. Tweet a is more valuable than Tweet b when $F(solution_a) < F(solution_b)$. Where F is the target function to be optimized, $solution_x$ is the math representation of Tweet x's content.

Hottest Tweet. Hottest Tweet is the optimal solution found by TO and its final value will be the result of TO.

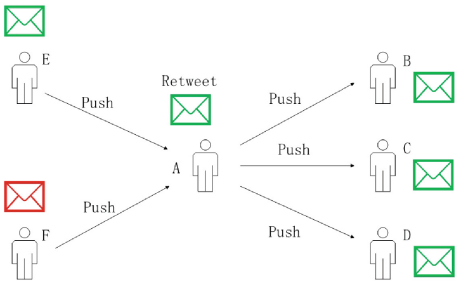
Following. Following is a kind of relationship on Twitter as shown in Fig. 1(a). If user A follows user B, A will be able to receive B's latest Tweet. This is similar to the swarm communication topology in SPSO [4]. But the difference is, TO has a dynamic topology which is continuously changing. Firstly, at the beginning of the algorithm, TO randomly initializes the following relationship of every user to generate a directed graph G. And each user is only allowed to

Algorithm 1. Twitter Optimization

procedure TWITTER
Initial M persons i.e. M Twitter users.
Randomly initial $Following_i$ for every person i where $Following_i$ is a set consists of the persons followed by person i.
Everyone randomly tweets and retweets.
Find the most valuable Tweet as the hottest Tweet.
for Each round **do**
for Each person i **do**
Receive the latest Tweet tweeted and retweeted by person p, where person p belongs to $Following_i$.
Choose the most valuable Tweet a and the least valuable Tweet b from what he received, then:
1. Retweets Tweet a and replace the hottest Tweet with a if a is more valuable.
2. Follow a's oral author (the source author of a who tweeted a first).
3. Unfollow b's oral author or retweeter.
if His own Tweet hasn't been retweeted for several rounds **then**
Tweet a new Tweet b and replace the hottest Tweet with b if b is more valuable.
end if
end for
end for
end procedure



(a) Following Relationship. (b) Tweeting.



(c) Choosing the best to Retweet.

Fig. 1. Behaviours on Twitter.

follow exactly F other users (F is a constant parameter). Then, in each round, each user will optimize his following set by following and unfollowing. He will choose the most valuable Tweet he received this round, and follow its oral author if he hasn't follow him yet. Once the following operation has been done, the user will unfollow a user in order to make his following set number remain to F. He will choose the least valuable Tweet and unfollow its sender. In other words, the following and unfollowing operation makes each user willing to follow the people who were spreading the valuable Tweet, i.e. better solution. Therefore, the whole topology becomes more efficient, and TO can make a fast-converging to the global optima.

Randomly tweeting for initialization. Tweeting means generating a new solution and sharing it to followers as shown in Fig. 1(b). During the initialization process of TO, each user randomly tweets to generate a random solution vector by the following Eq. (1):

$$x_i^k = min_i + (max_i - min_i) * random(-1, 1) \tag{1}$$

where x^k is the current solution of user k and x_i^k is the value of x^k in i-th dimension, min_i and max_i are bounds of solution value in i-th dimension. Note that the algorithm has no priori knowledge about the target function. The whole initialization process is totally stochastic.

Randomly retweeting for initialization. Every person k randomly chooses a person p from $following_k$ and retweets the Tweet of p for an initialization:

$$y_i^k = x_i^p \tag{2}$$

where y^k is the solution which user k is spreading. And x^p is the current solution found by user p.

Retweeting. Retweeting makes a user able to share the valuable Tweet to his followers as shown in Fig. 1(c). This behaviour can help the better solution spreading further. Besides, TO adds some updating steps during retweeting. When user k decides to retweets the Tweet of user p, user k randomly chooses one of the two operations below to update the Tweet he retweets.

Type 1 comment:

$$x_{i\ new}^p = x_i^p + Character^k * random(-1, 1) \tag{3}$$

Type 2 participation:

$$x_{i\ new}^p = x_i^k \tag{4}$$

When a user retweets a Tweet and chooses to *comment*, he will offer his idea about the Tweet. This idea can possibly improve the quality of the oral Tweet.

We use Eq. (3) to simulate this behaviour. After the Eq. (3) has been calculated for each dimension i, a better solution may be generated. x^p is the oral Tweet, i.e. solution, of user p and x^p_{new} is the new one. If $F(x^p_{new}) < F(x^p)$, the solution will be replaced by the new one. Meanwhile, *Character* is ranging from 10^{-6} to 1 randomly for each user, representing the different updating step sizes of different users.

When a user retweets a Tweet and chooses to *participate* in the related activity, he will contribute his own part to the activity. For example, the ice bucket challenge is an influential activity which is organized on Twitter. People from all over the world take up the challenge. We use Eq. (4) to simulate the phenomenon. Where we randomly choose only one dimension i and replace $x^p_{i\ new}$ with x^k_i, similar to the hybrid model in GA [5]. If $F(x^p_{new}) < F(x^p)$, the oral solution will be replaced by the new one.

Tweeting a new Tweet. This operation means discarding the original solution and regenerate a new one. At the start of each round, every user's follower number will be evaluated and everyone will have a corresponding label. If the follower number of a user ranks up to top 1 % of the whole crowd, he will be labeled as a *celebrity*. Otherwise, the user will be labeled as an *averageman*. For a *celebrity*, if his solution hasn't been updated during the last W times retweeting, he will regenerate a new one. Where W is the follower number of user k in the current round. For an *averageman*, he will regenerate a new solution every round. The *celebrity* will choose the most valuable Tweet whose author is an *averageman* as his new Tweet. And an *averageman* will regenerate a new solution by Eq. (5).

$$x^k_i = bestx_i + \delta * random(-1, 1) \tag{5}$$

Where *bestx* is the best solution ever found, δ is the exploring radius. Equation (5) generates a random solution near around the best solution. TO algorithm will finally stop when the number of iterations is greater than the upper bound N. For each iteration, the target function will be evaluated $O(M)$ times (tweet solution and retweet solution for every user to calculate fitness at most twice). So the total function evaluation number is $O(MN)$.

3 Experiment

3.1 Benchmark Functions for Comparison

In this section, eight benchmark functions are used to evaluate the performance of TO. Also, TO is compared with two typical methods, CPSO [6] and SPSO [4] (two variants of PSO) on the benchmark functions. The parameters of TO were set as: $F = 10$, $M = 30$, $N = evaluation\ number/60$, $\delta = 1$. And the parameters of CPSO and SPSO were set as [7]. The solution space in our experiment has 30 dimensions and the setting ranges of each dimension are [–100, 100]. Standard benchmark functions are listed in the Table 1.

Table 1. Eight benchmark functions to be tested

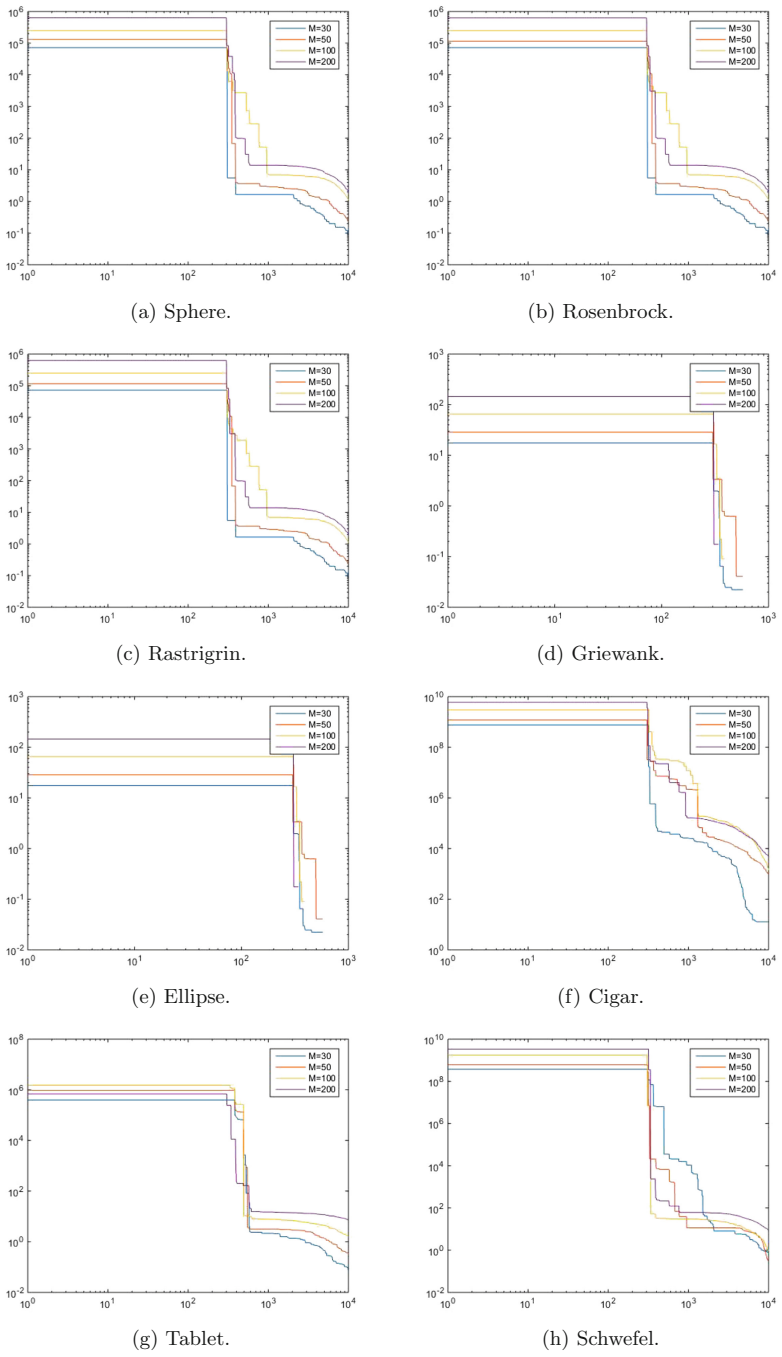
Function	Expression	Dimension	Optimum
Sphere	$F_1 = \sum_{i=1}^{D} x_i^2$	30	0
Rosenbrock	$F_2 = \sum_{i=1}^{D-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	0
Rastrigrin	$F_3 = \sum_{i=1}^{D}(x_i^2 - 10cos(2\pi x_i) + 10)$	30	0
Griewank	$F_4 = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} cos(\frac{x_i}{\sqrt{i}})$	30	0
Ellipse	$F_5 = \sum_{i=1}^{D} 10^4 \frac{i-1}{D-1} x_i^2$	30	0
Cigar	$F_6 = x_1^2 + \sum_{i=2}^{D} 10^4 x_i^2$	30	0
Tablet	$F_7 = 10^4 x_1^2 + \sum_{i=2}^{D} x_i^2$	30	0
Schwefel	$F_8 = \sum_{i=1}^{D}((x_1 - x_i^2)^2 + (x_i - 1)^2)^2$	30	0

Table 2. Statistical mean and standard deviation of solutions found by TO, CPSO and SPSO on eight benchmark functions over 20 independent runs

Function	Function evaluations	TO's Mean (StD)	CPSO's mean (StD)	SPSO's mean (StD)
Sphere	500000	0.000000	0.0000000	1.909960
		(0.000000)	(0.000000)	(2.594634)
Rosenbrock	600000	18.6452	33.403191	410.522522
		(6.9665)	(42.513450)	(529.389139)
Rastrigrin	500000	0.000000	0.053042	42.912843
		(0.000000)	(0.370687)	(2.177754)
Griewank	200000	0.000000	0.632403	2.177754
		(0.000000)	(0.327648)	(0.294225)
Ellipse	500000	0.000000	0.000000	53.718807
		(0.000000)	(0.000000)	(68.480173)
Cigar	600000	0.000000	0.000000	0.002492
		(0.000000)	(0.000000)	(0.005194)
Tablet	500000	0.000000	0.000000	1.462832
		(0.000000)	(0.000000)	(1.157021)
Schwefel	600000	0.000000	0.095099	0.335996
		(0.000000)	(0.376619)	(0.775270)

3.2 Parameter Selection

As shown in the Fig. 2, different values of M have different experiment results. When the population number M equals 30, TO gets it's best result on most of the functions.



(a) Sphere. (b) Rosenbrock.

(c) Rastrigrin. (d) Griewank.

(e) Ellipse. (f) Cigar.

(g) Tablet. (h) Schwefel.

Fig. 2. Different converging speed for different M. The horizontal axis stands for function evaluation times, the vertical axis stands for the best value found by TO and the optimum is zero. $F = 10$, $\delta = 1$, $N = 10000/60$.

3.3 Experiment Results and Analysis

As shown in the Table 2, TO obtains the best results among the three algorithms on the eight benchmark functions. $F1$, $F3$, and $F5$–$F8$ are unimodal functions. The results of these functions show the fast-converging ability of TO. $F2$ and $F4$ are multimodal functions with many local optima. The performance on the two functions shows the global search ability of TO in avoiding premature convergence.

All in all, TO has the following advantages: Firstly, TO's dynamic topology makes it hard to get stuck into a local convergence; Secondly, the adaptive updating step size can fit different solution spaces and different stages of the process; Thirdly, the replacement of one dimension helps the algorithm to jump out a local convergence especially when the problem is dimensionally independent; Finally, the multiple updating methods make TO able to handle different problems.

4 Conclusion

By imitating human's social behaviours on Twitter, an optimization algorithm named TO is proposed. Series of Twitter terms are defined as updating equations for exploration and exploitation. The experiment demonstrates that TO has a better convergence accuracy compared to CPSO and SPSO. It can be concluded that TO has a potential in solving optimization problems.

Acknowledgement. The authors would like to thank the anonymous reviewers for their time and valuable suggestions. This work is supported in part by the National Science Foundation of China under Grant Nos. (61375064, 61373001) and Jiangsu NSF grant (BK20131279).

References

1. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. Proc. IEEE Int. Conf. Neural Netw. **4**, 1942–1948 (1995)
2. Dorigo, M.: Optimization, learning and natural algorithms. PhD thesis. Politecnico di Milano, Italy (1992)
3. Dervis Karaboga, D.: An idea based on honey bee swarm for numerical optimization, Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
4. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: Proceedings of IEEE Swarm Intelligence Symposium, pp. 120–127 (2007)
5. Hassan, R., Cohanim, B., de Weck, O.: A comparison of particle swarm optimization and the genetic algorithm. Vanderplaats Research and Development (2005)
6. Tan, Y., Xiao, Z.M.: Clonal particle swarm optimization and its applications. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 2303–2309 (2007)
7. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13495-1_44