Journal of Computing and Electronic Information Management

ISSN: 2413-1660

A Verifiable Ions Motion Algorithm in Cloud Computing

Yingying Wu

Control Science and Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

wyy6796@163.com

Abstract: In this paper, we study the new type of ion motion algorithm optimization (IMO) validation of the problem, the algorithm simulation of anionic and cationic mutual attraction and repulsion to optimize, and outsourcing it to the cloud, which is to ensure that cloud according to the request execution IMO algorithm. A verifiable IMO algorithm and its verification algorithm are proposed to verify the authenticity and validity of the cloud.

Keywords: ion motion algorithm, optimization problems, cloud computing.

1. Introduction

In the past, numerous algorithms with inspiration from the nature have been proposed for solving various optimization problems. Some of the most popular are Genetic Algorithm (GA) [1–3], Differential Evolution Algorithm (DE) [4–7], Particle Swarm Optimization (PSO) [8], and Artificial Bee Colony (ABC) [9–11].When evaluating two algorithms, there are two conflicting standards: the speed of convergence and the quality of the final solution. Speed of convergence may lead to local optimal, on the other hand, the quality of the solution can lead to a broader search of the search space, resulting in slower convergence. To solve these two problems, the researchers improve the current algorithms or propose new techniques. In this paper, a new optimization algorithm called Ions Motion Optimization (IMO) is proposed as a competitive algorithm in this field.

The main inspirations of the IMO algorithm are two different ions: anion (a negative charged particle) and cation (a positive charged particle). The IMO algorithm divides the population of candidate solutions to two sets of negative charged ions and positive charged ions, and improve them according to the important characteristics of the ions "ions with the same charges repel each other, but with opposite charges attract each other"[12].

The principle and implementation of IMO is simple, but in practice it is often extremely complex. Because the real problems are sometimes extremely complex (e.g., multi-mode and high-dimensional).For this reason, when resources are limited by IMO algorithm has a complex issues to optimize, he can offload the computing to a third party with much more powerful computational resources. Especially in the coming age of cloud computing [13], a thin client such as smartphone can outsource this computing task to the cloud. There are great benefits of being doing so: the client can save the money on expensive hardware for running the algorithm; and he can get the desirable solution from the cloud with the help of outsourcing.

However, there are security and privacy challenges in cloud computing environments [14], [15]. Although outsourcing the running of IMO algorithm has attractive superiority, it induces a serious problem: how could the result returned by the cloud be verified? In other words, to reduce overhead cost, the cloud could possibly cheat the client by not faithfully running the IMO algorithm, or even by returning a random guess without executing the algorithm at all.

In this article, we believe that the outsourcing of IMO algorithms and the first validation are our best practices. In contrast to existing verifiable computations based on expensive encryption techniques, we propose a simple and effective solution. To simplify the problem, we made some reasonable assumptions that customers could verify the authenticity of the cloud.

2. The Problem Definition

We believe that a client has a complex problem that F needs to be optimized by IMO. Then he outsourced the IMO algorithm to a cloud with strong computing power. After the cloud solves the problem, by running the IMO algorithm, it returns the result to the client. When the client receives the result from the cloud, it wants to verify that the returned result is correct. In other words, whether the cloud has invoked the IMO algorithm faithfully.

To simplify the problem, we made some assumptions about the IMO algorithm. First of all, let's assume that F is a minimum optimization problem. Second, determine the parameters. To simplify the problem, we only consider that the client can configure the stop criteria because the other parameters are not strictly a problem and are usually recommended. Finally, the stop standard of IMO algorithm: when the largest iteration number t stops running.

Based on these hypotheses, we can model the IMO outsourcing problem as

$$\mathrm{r} = \mathrm{IMO\_outsouce}(t)$$

where $\mathrm{IMO\_outsouce}(*)$ is the IMO algorithm to be outsourced to the cloud for the optimizing problem F; t is a parameter whose value is provided by the client; r is the result returned by the cloud, and it is to be verified by the client.

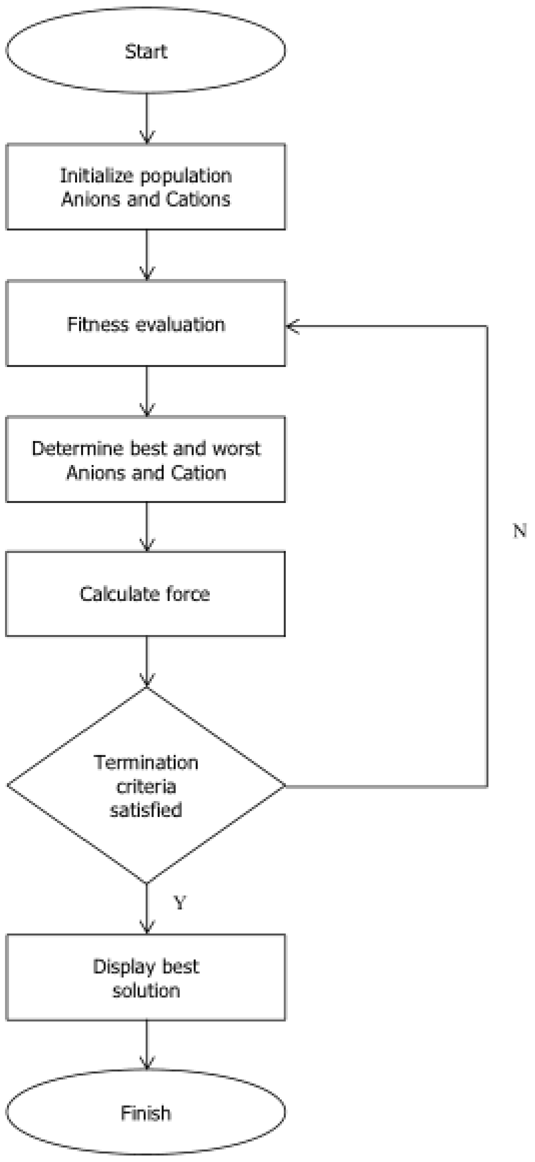
Start
Initialize population Anions and Cations
Fitness evaluation
Determine best and worst Anions and Cation
Calculate force
Termination criteria satisfied
N
Y
Display best solution
Finish

Fig.1

3. IMO Algorithm Implementation

Ions have more freedom to move. In addition, attraction forces among ions with opposite charges are more than repulsion force among ions with similar charges [20]. We require IMO to ignore repulsion forces in this phase in order to explore the search space. We assume that the only factor for computing attraction force is the distance between ions. The concrete implementation process is as follows: first, generate the initial population. By using RANDOM function, the system randomly generates S, which satisfies the condition, namely, the population size is S. Then, we substituted the target function F to calculate the corresponding adaptive value of each ion, and compared the adaptive value to determine the best cation and the worst anion (the adaptive value smaller the better). Calculate the attraction between ions and establish the mathematical model as follows

$$AF_{i,j} = \frac{1}{1+e^{-0.1/AD_{i,j}}}, CF_{i.j} = \frac{1}{1+e^{-0.1/CD_{i,j}}}$$

$$AD_{i,j} = |A_{i,j} - Cbest_j|, CD_{i.j} = |D_{i,j} - Abest_j|$$

update the positions of anion and cation

$$A_{i,j} = A_{i,j} + AF_{i,j} \times (Cbest_j - A_{i,j}), C_{i,j} = C_{i.j} + CF_{i,j} \times (Abest_j - C_{i,j})$$

where i is the ion index, j indicates dimension, $AD_{i,j}$ is the distance of $\mathrm{i-th}$ anion from the best cation in$\mathrm{j-th}$ dimension, $CD_{i.j}$ calculates the distance of $\mathrm{i-th}$ cation from the best anion in $\mathrm{j-th}$ dimension, $AF_{i,j}$ is the resultant attraction force of anions, and $CF_{i.j}$indicates the resultant attraction force of cations.

Finally, determine whether the termination condition is satisfied: do you reach the maximum iteration value t?If the maximum iteration value is reached, the output optimal solution, if not achieved, will continue to iterate.

The general steps of the IMO algorithm are presented in Fig.1.

4. The Proposed Scheme

A. General Framework

The core of our proposed scheme is a verifiable IMO algorithm. In order to solve the formulated problem, there are three phases to fulfill the outsourcing of the proposed algorithm and its verification.

Request: The client sends a outsourcing request to the cloud. The request contains the verifiable IMO algorithm and its arguments including additional information for subsequent verification.

Outsourcing Computing: After receiving the outsourcing request from the client, the cloud can verify the IMO algorithm with parameters running.And finally returns the output to the client.

Verification: When the client receives a reply from the cloud, run the validation algorithm to verify that the cloud is honest.

B. Verifiable IMO Algorithm

We now describe the verifiable IMO algorithm in detail. The algorithm and its arguments are provided by the client and it is run by the cloud.

The input arguments include the maximum iteration number t, the signature $\mathrm{t\_sig} = E_{sk}(t)$ of t by the user's private key sk , and the public key pk corresponding to sk. The output of the algorithm consists of a status value indicating whether the arguments from the client are correctly passed in, the optimized solution, and the public key pk. To further ensure the indistinguishability of status value for the cloud and the privacy for the client, all these values are encrypted by pk before being output. N

In the algorithm, we first need to verify the arguments to check whether the value of t is really what the client requested. This can be done by verifying the signature of t, i.e. by decrypting $\mathrm{t\_sig}$ with $\mathrm{pk}\left(D_{pk}(\mathrm{t\_sig})\right)$ and checking whether the decrypted value is identical with t. If the arguments verification succeeds, we then follow the procedures of traditional IMO algorithm to find the optimized solution G, and return the result $\mathrm{r} = E_{pk}(ture, G, pk)$.

C. Verification Algorithm

When the client receives response from the cloud, he runs the verification algorithm to verify the result. As the result r is returned in encrypted format, the client needs to decrypt it using his private key sk. After that, he verifies the following points:

1) the returned public key is unchanged; 2) the arguments verification succeeds; 3) the problem F is really optimized by IMO algorithm.

The first two points are quit easy to be accomplished. If the private key of the client can successfully decrypt the returned result, the public key is not changed.And if the cloud invokes the verifiable IMO algorithm with the arguments from the client, i.e. t, $\mathrm{t\_sig}$, and pk, the arguments verification will succeed since the decryption of signature $\mathrm{t\_sig}$ with pk is t exactly. But the last one is a challenge because the solution found by IMO algorithm is suboptimal and it may be different from the real global minimum. For this reason, it is hard to find a deterministic method to verify the solution in general. According to this difficulty, this paper proposes a simple and effective method to verify this solution.The basic idea is to disrupt all the solutions received and verify that they are optimal. The client generates n disturbed values of G within the range of $[-\delta, \delta]$. If these n solutions are better than the cloud's response G by more than ϵ in a ratio greater than θ, then the verification fails because the cloud may cheat in extremely high probability.

5. The Experimental Analysis

Five benchmarks being widely used in literature are adopted in our experiments, and their mathematical representations and related parameters configuration are listed in Table I. The population size is set to 30 throughout the experiments.

The experimental part is verifying that if the cloud returns the optimal solution of a random guess.t is set to 5000, and the random flip of coin b is used to simulate whether the cloud is cheating.If $\mathrm{b} = 0$, then run the verifiable IMO algorithm and output the result, otherwise it will generate a random number of guesses as the output. All output results will be validated, and each of these benchmark functions will be repeated 5000 times.The accuracy of the final verification results is above 99.9%, which indicates the feasibility of the scheme.

The verification results are shown in Table 2. It is found that for all benchmark functions, the correct verification ratios are greater than 99.9%, which demonstrates the feasibility of our scheme.

Table 1 Benchmark Functions

Function	Name	Mathematical representation	Dimension	Domain	Global Minimum
f_1	Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]$	0
f_2	Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} ((100x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	$[-30,30]$	0
f_3	Rastrigrin	$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$[-5.12,5.12]$	0
f_4	Griewank	$f_4(x) = \left(\frac{1}{4000}\right)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\frac{x_i}{\sqrt{i}}\ ) + 1$	30	$[-600,600]$	0
f_5	Ackley	$f_5(x) = -20exp - 0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	$[-32,32]$	0

Table 2 Experimental Results

Function	f_1	f_2	f_3	f_4	f_5
Correct	99.92%	99.97%	99.91%	99.90%	99.95%
False positive	0.08%	0.00%	0.01%	0.09%	0.00%
False negative	0.00%	0.03%	0.08%	0.01%	0.05%

6. Conclusion

To the best of our knowledge, for the first time, we study the outsourcing of IMO algorithm and its verification. Simple and effective, unlike ordinary verifiable computing schemes rely on expensive encryption.

References

[1] D. Beasley, D. Bull, R. Martin, An Overview of Genetic Algorithms. Part 2, University of Cardiff, Cardiff, 1993.

[2] D. Beasley, D. Bull, R. Martin, An Overview of Genetic Algorithms. Part 1, University of Cardiff, Cardiff, 1993.
[3] D.B. Fogel, What is evolutionary computation, in: IEEE Spectrum, 2000, pp.26–32.
[4] K. Price, R. Storn, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997)341–359.
[5] K.Price, R. Storn, Differential Evolution a Practical Approach to Global Optimization, Springer Natural Computing Series, 2005.
[6] P. Bergey, C. Ragsdale, Modified differential evolution: a greedy random strategy for genetic recombination, Omega 33 (2005) 255–265.
[7] A.Salman, A. Engelbrecht, M. Omran, Empirical analysis of self-adaptive differ-
ential evolution, Eur. J. Oper. Res. 183 (2007) 785–804.
[8] O.M. Nezami, A. Bahrampour, P. Jamshidlou, Dynamic Diversity Enhancement in Particle Swarm Optimization (DDEPSO) algorithm for preventing from premature convergence, Procedia Comput. Sci. 24 (2013)54–65.
[9] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for realparameter optimization, in: Information Sciences, 2012, pp. 120–142.
[10] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, in: Applied Soft Computing, 2008, pp. 687–697.
[11] W. Gu, M. Yin, C. Wang, Self adaptive artificial bee colony for global numerical optimization, IERI Procedia 1 (2012) 59–65.
[12] M.S. Silberberg, Principles of General Chemistry, McGraw-Hill, New York, 2007.
[13] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A.Rabkin, I.Stoica, and M.Zaharia,"A view of cloud computing," Communications of the ACM, vol. 53,no. 4, pp. 50–58, 2010.
[14] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security Privacy, vol. 8,no. 6, pp. 24–31, 2010.
[15] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," IEEE Communications Surveys Tutorials, vol. 15, no. 2, pp. 843–859, 2013.