International Journal of Pure and Applied Mathematics
Volume 106 No. 1 2016, 301-306
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v106i1.24

GLOBAL OPTIMIZATION OF FUNCTIONS OF SEVERAL VARIABLES USING PARALLEL TECHNOLOGIES

Igor Grigoryev[1 §], Svetlana Mustafina[2]
[1,2]Sterlitamak Branch of the Bashkir State University
37, Lenin Avenue, Sterlitamak city, 453103, Russia;

Abstract: In this paper, on the basis of the method particle swarm optimization was developed the algorithm of the parallel search of global extremum. In the system of parallel programming on C language implemented method of particle swarm for the global minimization of functions. The performance of the parallel method was tested for two famous benchmark optimization problems (Styblinski Tang function and Rastrigin function) and compared with the results obtained by employing the sequential method. Conducted research on the effectiveness of parallelization has shown the advantage of the parallel algorithm.

AMS Subject Classification: 65Y05, 68W10, 52A40
Key Words: global extremum, the method of particle swarm, parallel computing, Nvidia CUDA

1. Introduction

Copying the actions of nature, man creates more and more sophisticated optimization algorithms. To create them often are examples from nature, such as the genetic code, or bird behavior modeling migration of fish, cooling of the metal, etc [1]. Currently in production and business optimization algorithms are widely used because they provide an opportunity to save not only money, but also time, which is not always enough.

Received: December 7, 2015
Published: February 5, 2016

§Correspondence author

2. Theoretical Part

A kind of real optimization problem can be formulated as the following functional optimization problem.

$$\min_{x \in R^{\mathsf{n}}} f(x) = f(x^*), x = (x_1, x_2, \ldots, x_n)\,, \tag{1}$$

where $f(x)$ is the objective function, and x is the decision vector with n decision variables.

To solve this problem was chosen the method of particle swarm optimization (PSO). Particle swarm optimization is a heuristic global optimization method put forward originally by Eberhart R and Kennedy J in 1995 [2]. It is developed from swarm intelligence and is based on the research of bird and fish flock movement behavior. While searching for food, the birds are either scattered or go together before they locate the place where they can find the food. While the birds are searching for food from one place to another, there is always a bird that can smell the food very well, that is, the bird is perceptible of the place where the food can be found, having the better food resource information. Because they are transmitting the information, especially the good information at any time while searching the food from one place to another, conduced by the good information, the birds will eventually flock to the place where food can be found. As far as particle swam optimization algorithm is concerned, solution swam is compared to the bird swarm, the birds moving from one place to another is equal to the development of the solution swarm, good information is equal to the most optimist solution, and the food resource is equal to the most optimist solution during the whole course. The most optimist solution can be worked out in particle swarm optimization algorithm by the cooperation of each individual. The particle without quality and volume serves as each individual, and the simple behavioral pattern is regulated for each particle to show the complexity of the whole particle swarm.

Due to its many advantages including its simplicity and easy implementation, the algorithm can be used widely in the fields such as function optimization, the model classification, machine study, neutral network training, the signal procession, vague system control, automatic adaptation control and etc.

3. An Algorithm for Solving Global Optimization Problem

PSO is a population-based optimization technique proposed firstly for the above unconstrained minimization problems. In a PSO system, multiple candidate solutions coexist and collaborate simultaneously. Each solution called a particle, flies in the problem search space looking for the optimal position to land. A particle, as time passes through its quest, adjusts its position according to its own experience as well as the experience of neighboring particles. Tracking and memorizing the best position encountered build particles experience. For that reason, PSO possesses a memory (i.e. every particle remembers the best position it reached during the past). PSO system combines local search method (through self experience) with global search methods (through neighboring experience), attempting to balance exploration and exploitation.

A particle status on the search space is characterized by two factors: its position and velocity, which are updated by following equations.

$$\vec{v} \leftarrow \omega \cdot \vec{v} + C_1 \cdot rnd() \cdot (\vec{p}_{best} - \vec{x}) + C_2 \cdot Rnd() \cdot (\vec{g}_{best} - \vec{x}), \tag{2}$$

$$\vec{x} \leftarrow \vec{x} + \vec{v}, \tag{3}$$

where $\vec{v}$ called the velocity for particle, which represents the distance to be traveled by this particle from its current position; $\vec{x}$ represents the position of particle; the coefficient ω, named Juha Shi (Yuhui Shi) and Russell Eberhard the inertia weight, that controls the impact of previous velocity of particle on its current one; C_1, C_2 - are positive constant parameters called acceleration coefficients; $\vec{p}_{best}$ - represents the best previous position of particle(i.e. local-best position or its experience); $\vec{g}_{best}$- represents the best position among all particles in the population (i.e. global-best position); $rnd()$ and $Rnd()$ are two independently uniformly distributed random variables with range $[0, 1]$.

In PSO, (2) is used to calculate the new velocity according to its previous velocity and to the distance of its current position from both its own best historical position and its neighbors best position. Generally, the value of each component in can be clamped to the range to control excessive roaming of particles outside the search space. Then the particle flies toward a new position according (3). This process is repeated until a termination criterion is reached.

4. Results of Computational Experiment

In the proposed algorithm, it is assumed for computation, the central processing unit (CPU) and graphics processing unit (GPU), i.e., the so-called heteroge-

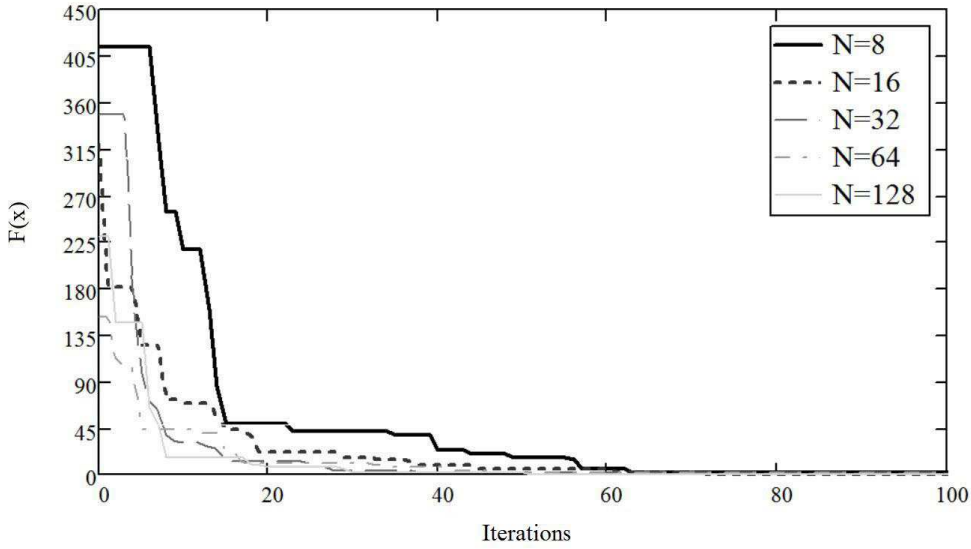

Figure 1: The rate of convergence of the method according to the amount of particles in the swarm for function Rastrigin (n = 3)

neous computing environment. Computations that can be performed independently are performed on the GPU. Areas that can not be parallelized will be operated by CPU. Run parallel sections of code on the graphics device allow technology CUDA [3].

On the basis of developed algorithm implemented the program, approved on Styblinski Tang function and Rastrigin function. Fig.1 shows the results of the particle swarm in the method depending on the number of particles in the swarm (8, 16, 32, 64 and 128 particles) in three-dimensional space.

From Fig.1 shows how the accuracy of the solution increases with an increase in the swarm, but at the same time increases significantly during operation. In this regard, it was decided to to implement parallel algorithm for nVidia Cuda.

Most of the known methods of particle swarm are sequential. Parallel methods are little known, and they all appeared after 2004. In this paper presents a method of using the island model of concurrency. The basic idea of this method is as follows: the whole swarm of particles N is divisible by m Islands (number of computing devices in the system) and the particles belonging to each of the islands, are processed on your graphics processor. After each iteration k independent islands are sharing best particles.

To compare sequential and parallel implementations of the algorithm of

Table 1: Comparative analysis of parallel and sequential algorithms for Styblinski Tang function.

n	N	T_{CPU}	T_{GPU}	Acceleration
2	64	1,269	1,262	1,006
	128	1,299	1,256	1,034
	256	1,330	1,256	1,059
3	256	1,350	1,272	1,061
	1024	1,607	1,329	1,209
	4096	2,659	1,483	1,792
4	256	1,416	1,316	1,076
	1024	1,731	1,352	1,280
	4096	3,174	1,591	1,994

Table 2: Comparative analysis of parallel and sequential algorithms for Rastrigin function

n	N	T_{CPU}	T_{GPU}	Acceleration
2	64	1,601	1,377	1,163
	128	1,794	1,418	1,256
	256	2,226	1,461	1,524
3	256	2,558	1,512	1,692
	1024	6,069	1,963	3,092
	4096	20,237	3,567	5,673
4	1024	8,350	2,253	3,706
	2048	13,926	3,120	4,463
	4096	27,673	4,326	6,397

particle swarm, a series of experiments. Testing was conducted on algorithms computing system with a graphical computing device GeForce GT 520M, CPU Intel Core i5-2410M 2.3 GHz, the operating system Microsoft Windows 7 driver installed NVIDIA CUDA Version 5.5. Each program starts about 10 times, was taken as a result of the arithmetic mean of the data (see Table 1, Table 2).

5. Conclusion

As seen on the CPU time spent on finding an extremum more. It should be noted that the computational experiment on the GPU is also taken into account the time required to allocate memory on the graphics processor and then copy it to the original data. Conducted research on the effectiveness of parallelization has shown the advantage of the parallel algorithm.

References

[1] T. Weise, *Global Optimization Algorithms – Theory and Application*, Germany (2009).

[2] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of IEEE International conference on Neural Networks*, **4** (1995), 1942 - 1948.

[3] Grigoryev I.V., Mustafina S.A. Algorithm of global optimization of functions with use of parallel technologies, *Journal "Science Bulletin"*, **2**, No. 2 (2014), 145-153.