Parallel evolutionary algorithm in high-dimensional optimization problem

S. N. Kotriakhova[a], M. M. Stepanova[b]

Saint-Petersburg State University,
1, Ulyanovskaya str., Saint Petersburg, 198504, Russia

E-mail: [a] dangemy@gmail.com, [b] mstep@mms.nw.ru

An implementation of the combined evolutionary algorithm for searching extremum of functions with many parameters is proposed. The algorithm designed to optimize parameters of the molecular-dynamics reactive force field potential ReaxFF also can be efficient in many other extrema-searching problems with arbitrary complex objective function. The algorithm itself is a hybrid of two evolutionary methods: Genetic Algorithm which uses the principle of natural selection in the population of individuals; and Particle Swarm Optimization which imitates the self-organization of the particle swarm. Individuals in population as far as swarm's particles are treated can be considered as trial solution's vectors. Combination of these two methods provides an opportunity to work with objective functions with unknown complex structure which often has a composition of specific peculiarities insurmountable by simple algorithms.

Genetic Algorithm parameterization regarding choosing its main strategies for computations with different objective functions has been analyzed. Results for classical test functions convergence speed testing presented. Effectiveness of the algorithm working at the computational system with shared memory and at analogous distributed system has been compared and good scalability of implemented algorithm at distributed computational system demonstrated.

Keywords: numerical simulation, optimization algorithms, genetic algorithm, parallel algorithm, absolute extremum search

Introduction

Optimization of the function describing the real world objects and phenomena can face some specific peculiarities and nuances. This work aimed to design the algorithm to work effectively with the high-dimensional complex landscape functions of high computational complexity. To deal with such difficulties, parallel algorithm based on evolutionary optimization methods has been implemented. Moreover, the algorithm is universal what allows to use it with a wide variety of problems

This algorithm is developed as an alternative to our MGSA [Stepanova, Shefov, …, 2016] for optimization (global search) of the reactive force field (ReaxFF) parameters [Nomura, Kalia, …, 2008]. This empirical force field potential is implemented in some molecular dynamics simulators and allows simulating chemically reactive systems. But to simulate the particular system one should solve a very complex computational problem by choosing numerous (more than hundred) potential parameters. To do so, objective function should be constructed. Objective function depends on these parameters and usually is constructed as a weighted sum of squared differences of set of characteristics for simple compounds calculated with quantum methods (reference) and with ReaxFF. This sum allows estimating the difference between the quality of models with chosen ReaxFF parameters and reference models. Then such multivariable function should be minimized. Resulting minimized set of parameters can be used to simulate large-scale system of analogical compounds.

Some important peculiarities of objective function are essential for choosing the method for problem solution. Firstly, high-dimensional objective function complicates convergence to the global optimum and makes some algorithms almost useless. Secondly, the landscape of an objective function is unknown and can have peculiarities of arbitrary type. Particularly, analysis of the specific objective function landscape for Zn-O-H compounds shows two features: "canyons' with a narrow valley which is easy to find, but difficult to converge to the minimum; and "needle in the hay", where the global minimum has a small area relative to the search space. Furthermore, the "needle" can be placed at the bottom of the "canyon". And last, at each step of optimization for each model of optimizing set the molecular dynamics simulation should be done. This operation is highly resource-intensive because of using the MD-simulator LAMMPS for each calculation.

Hybrid evolutionary algorithm

The hybrid algorithm designed to solve the abovementioned problem combines two evolutionary methods of optimization. The first one is the genetic algorithm (GA) – heuristic optimization method inspired by the natural selection [Mitchell, 1999]. This method allows dealing with objective functions of any dimension without any requirements to the function's properties. Also, GA with a properly chosen parameters avoids the problem of premature convergence to the local optimum. The main idea of GA is the evolution of the population taken as a set of probe solutions (vectors). The initial population is formed randomly, afterwards the value of the objective function is calculated for each sample. If none of them satisfies the convergence conditions (not an optimum with given precision), a new population is formed: pairs of parents selected from previous population are breeding and mutate to obtain the children; objective function for children is also calculated, and the best samples are selected into the new population. New populations formation is continued while the convergence criterion doesn't meet: either the optimum will be reached, or the maximum number of iteration.

The second part of the hybrid algorithm is the particle swarm optimization (PSO) – stochastic optimization method based on the idea of self-organization of the particle swarm [Kennedy, Eberhart, 1995]. The population means here the particle swarm and each particle is described by its coordinate and velocity. The key to convergence is the ability of swarm to follow the best particle.

In hybrid algorithm approach the set of probe solutions for a given number of iterations is treated either as GA population or as PSO particle swarm. GA provides the global search for areas expected to

contain the minimum, and PSO localizes the minimum precisely. Thereby, the hybrid algorithm can effectively optimize multimodal high-dimensional functions with complex landscape.

Algorithm performance check and choice of parameters

Test functions such as de Jong, Michalewicz, Rastrigin, Schwefel and Rosenbrock functions are useful to evaluate characteristics of optimization algorithms. For every test function it is possible to choose highly specialized algorithm which gives the best results in respect to the rate of convergence and precision. However, searching such algorithm requires some time. For the real world problem it can be inexpedient. This algorithm universally tackles different problems without additional fine tuning. At the same time, algorithm's implementation allows to optimize parameters of the algorithm itself to speed up the convergence on given function. For example, by optimizing the number of consequent iterations of GA and PSO for Michalewicz function (Fig.1, a):

$$f(x) = -\sum_{i=1}^{n} \sin(x_i) \cdot \left(\sin\left(\frac{i \cdot x_i^2}{\pi} \right) \right)^{2m},$$

where $0 \le x_i \le \pi, i \in [1; n]$ and minimum is $f(x^*) = -1.8013, x^* = (2.20, 1.57), n = 2, m = 10$, it was found the ratio which gives the best rate of convergence – with the minimum reached after 439 iterations (25 iterations of GA and 20 iterations of PSO alternately).

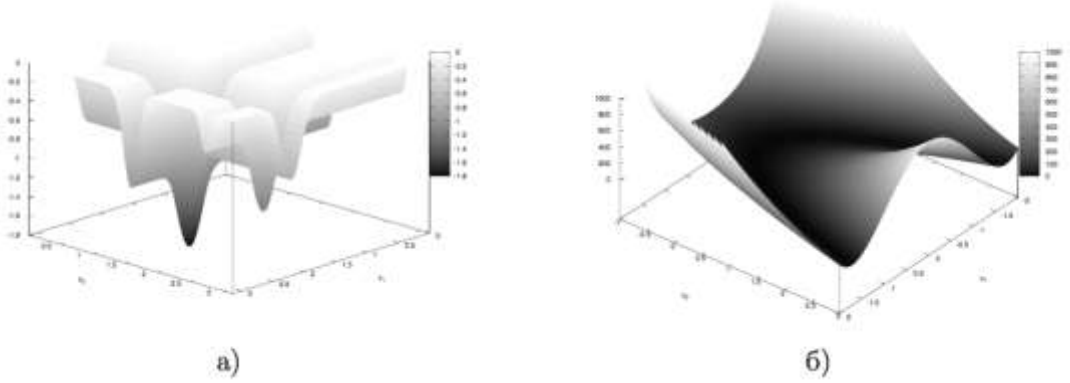

Figure 1. a) Michalewicz function for n = 2; b) Rosenbrock function for n = 2

Genetic algorithm implementation presented in this work allows to choose different operators of mutation, breeding, parents selection and selection to the new population. To optimize the ReaxFF potential the following methods have been used: phenotype outbreeding to choose the parents - the most distant from each other (Euclidean metric) candidate solutions; breeding using single-point crossover – each child gets one part of the genes (vector's components) from one parent and the other part from another; only descendants whose parents had identical genes are mutated; elitism for selection – best candidate solutions of all parents and children goes to the new population.

In this configuration the algorithm was tested for different test functions with evaluation of the influence of population size on the rate of convergence. Table 1 shows the result for Rosenbrock function:

$$f(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right],$$

where $-5 \le x_i \le 10, i \in [1; n]$, and minimum $f(x^*) = 0, x^* = (1, ..., 1)$. This is an unimodal function with the global minimum lying in a narrow, parabolic "canyon" (Fig. 1, b). Similar peculiarity was

found during analysis of the ReaxFF potential objective function. Here, in one iteration $2 \times \text{size of population}$ vectors are processed, so the number of processed vectors should be compared instead of the number of iterations. The table shows that the bigger population size leads of a higher rate of convergence.

Table 1. Rate of convergence for different population size for Rosenbrock function with precision 0.0001, n = 10

Population size	Average number of iterations	Number of processed vectors
8	203811	3260976
16	79509	2544288
32	30831	1973184
64	11140	1435920
128	5351	1369856
256	2606	1334272

Parallel implementation of the algorithm

To calculate the value of the objective function it is necessary to find the values of the charac-teristics of simple models (potential energies and forces components). To perform such calculations we are using the MD-simulator LAMMPS. Due to a significant amount of time required for this calculations and the ability to perform them independently for each vector of population the corresponding part of the algorithm has been implemented as parallel using the MPI (OpenMPI implementation). Substantial benefit from parallelism is provided by the absence of interprocess exchange in parallel section. At the beginning of each iteration Master-process conducts GA (parents selection, breeding, mutation) or PSO (coordinates and velocities updating) operations; then the population array splits into subarrays, which are sent to all processes. All processes compute the values of objective function for all vectors from its own subarray, after that Master-process gathers this values and matches them to corresponding vectors (Fig. 2).

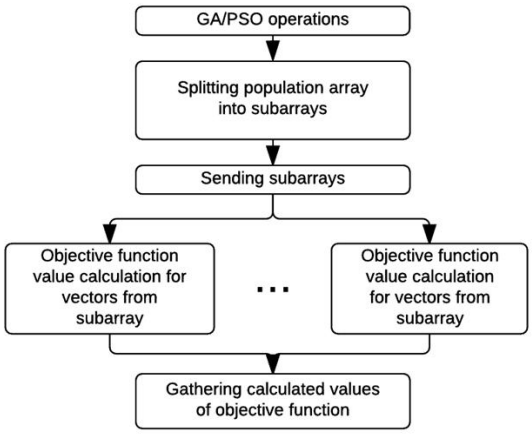

Figure 2. Flowchart of the parallel section

Fig. 3 represents the comparison of algorithm's speed on distributed systems and shared-memory systems. There were used 4 nodes based on Intel Core i5 CPU 3.00 GHz, RAM 8 Gb connected through 1 Gb Ethernet. Parallel implementation provides noticeable increase of algorithm's speed. The absence of exchange allows to use distributed systems with a large number of nodes. It should be noted, that the greater time for the shared-memory system with 2 and 4 processes is caused by a need of writing to a disk input files for LAMMPS and, respectively, waiting for access to shared disk.

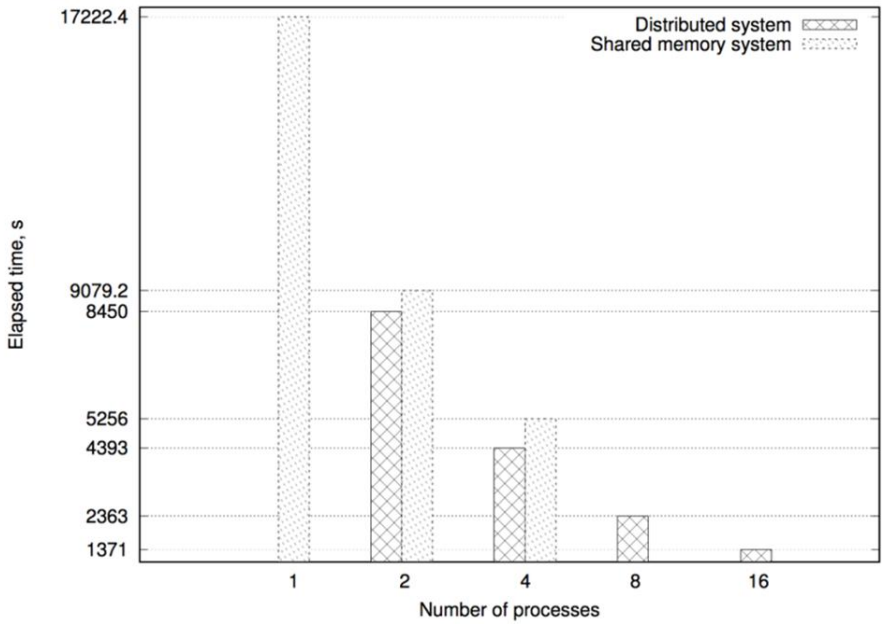

Figure 3. Calculation time for different numbers of processes

Optimization results for the ReaxFF potential

The set of optimized parameters has been obtained for the Zn-O-H system in the case of the original problem of optimization the parameters of potential ReaxFF. This set provides good results in case of molecular dynamics simulations. Optimization was performed for 24 most significant parameters. With population size equal to 128 vectors the result was achieved at 3361 iterations, which took 179 hours with 4 processes at Intel Core i5 CPU 3.00 GHz, RAM 8 Gb. With comparable quality of the result the presented algorithm is much faster than MGSA [Stepanova, Shefov, …, 2016].

Conclusion

The designed original hybrid evolutionary algorithm implementation effectively tackles high-dimensional functions optimization. Algorithm is universal and can deal with different objective functions, even with a complex fitness landscape, as it has been demonstrated at test functions. Besides, the fine tuning of the algorithm for optimal work with specific function can be done via the algorithm itself. Parallel implementation significantly increases the speed of work in case of computationally intensive object functions. Hybrid algorithm demonstrates good results while optimizing the MD-potential ReaxFF.

References

Nomura K., Kalia R. K., Nakano A., Vashishta P. A scalable parallel algorithm for large-scale reactive force-field molecular dynamics simulations // Computer Physics Communications. — 2008. — Vol. 178. — P. 73–87.

Stepanova M.M., Shefov K.S., Slavyanov S.Yu. Multifactorial global search algorithm in the problem of optimizing a reactive force field // Theoretical and Mathematical Physics. — 2016. — Vol. 187, Issue 1. — P. 603—617.

Mitchell M. An Introduction to Genetic Algorithms // Cambridge, MA, USA: MIT Press. — 1999. — 221 p.

Kennedy J., Eberhart R. Particle swarm optimization // Piscataway, NJ: Proceedings of IEEE International Conference on Neural Networks. — 1995. — P. 1942–1948.