Local Tuning in Multilevel Scheme of Parallel Global Optimization

Konstantin Barkalov[1,a)] and Ilya Lebedev[1]

[1] *University of Nizhni Novgorod, Nizhni Novgorod, Russia.*

[a)]Corresponding author: barkalov@vmk.unn.ru

Abstract. This work considers a parallel algorithm for solving the multidimensional multiextremal optimization problems. The algorithm combines nested optimization scheme and Peano-type space filling curves for the dimensionality reduction. To decrease the number of iterations of the global algorithm, we use one of local tuning techniques based on the adaptive estimation of the global optimizer. Parallel algorithm with mixed local-global strategy of search is proposed as well. The efficiency of the parallel algorithm was investigated using a supercomputer. The speedup of the algorithm using several nodes as compared with the serial algorithm has been demonstrated experimentally.

INTRODUCTION

This paper considers the problem of multidimensional multiextremal optimization

$$\phi(y^*) = \min\{\phi(y) : y \in D\}$$

$$D = \{y \in R^N : a_i \leqslant x_i \leqslant b_i, 1 \leqslant i \leqslant N\}$$

where $a, b \in R^N$ are given vectors. As for the class of the considered problems, the fulfillment of the two following important conditions is presupposed.

First, it is supposed that the objective function $\phi(y)$ can be defined non-analytically, but using an algorithm for calculation of its values at some points of domain D. In this case, such a calculation (*trial*) is computationally intensive.

Second, we shall assume that $\phi(y)$ satisfies Lipschitz condition

$$|\phi(y_1) - \phi(y_2)| \leqslant L\|y_1 - y_2\|, y_1, y_2 \in D, 0 < L < \infty$$

that is typical for the applied problems [1, 2, 3, 4].

Solving the multiextremal optimization problems is significantly more computationally intensive as compared with other types of optimization problems, since the global optimum is an integral characteristic of the problem solved and requires an analysis of the whole search domain. The efficient algorithms for solving the problems of this kind have been developed in Lobachevsky State University of Nizhnii Novgorod within the framework of the information-statistical approach [5, 6]. At present, the parallelization of the algorithms is the main direction of further development of these ones since the use of the supercomputer systems and of the parallel computing technologies allows expanding the field of application of the global optimization algorithms essentially [7, 8, 9, 10, 11, 12].

This paper continues the series of studies the initial results of which were provided in [13, 14].

PARALLEL ALGORITHM FOR SOLVING THE MULTIDIMENSIONAL PROBLEMS

DIMENSION REDUCTION

There is a number of ways to adapt effective one-dimensional algorithms for solving multidimensional problems; see, for example, the diagonal partitions method in [15, 16] or the simplicial partitions method in [17].

In the present study, we will use a mixed scheme of the dimension reduction utilizing Peano curve and the nested optimization scheme. Such schemes (used separately) allow reducing the solving of the multidimensional optimization problems to the solving of one or several linked one-dimensional optimization problems.

The use of Peano-type space filling curves (*evolvents*) $y(x)$, which map a unit interval $[0, 1]$ onto an n-dimensional cube continuously and unambiguously allows the reduction of the problem of minimization in domain D to a minimization problem on the interval $[0, 1]$

$$\phi(y^*) = \phi(y(x^*)) = \min\{\phi(y(x)) : x \in [0, 1]\} \tag{1}$$

The second method for the dimension reduction is the multilevel scheme of nested optimization based on the relation

$$\min_{y \in D} \phi(y) = \min_{u_1 \in D_1} \min_{u_2 \in D_2} \dots \min_{u_M \in D_M} \phi(y) \tag{2}$$

where the i^{th} variable ui is a vector of the dimensionality N_i of the components of vector y taken sequentially i. e.

$$y = (y_1, y_2, \dots, y_N) = (u_1, u_2, \dots, u_M), \tag{3}$$

In scheme (2), the nested subproblems are the multidimensional ones, and the dimension reduction method based on Peano curves can be applied for solving of these ones. The number of vectors u_i and the quantity of components in each vector are the parameters of the block recursion scheme and can be utilized for the forming of the subproblems with necessary properties. For example, if $M = N$ i. e. $u_i = u_i, 1 \leqslant i \leqslant N$, then each nested subproblem is a one-dimesional one. If $M = 1$ i. e. $u = u_1 = y$, then solving the problem is an equivalent to its solving using a single evolvent mapping $[0, 1]$ onto D; the nested subproblems are absent. The issues of the numerical construction of the Peano curve approximations and of using the nested optimization scheme are considered in details in [3, 4].

PARALLEL GLOBAL SEARCH ALGORITHM

To solve the one-dimensional subproblems in the nested optimization scheme, we will use characteristical global search algorithm (GSA). The first iteration x^1 is executed in an arbitrary internal point of the search interval. The choice of the next iteration pointvx^{k+1}, $k \geqslant 1$ is performed according to the following rules.

1. To arrange the search information in the order of increasing values of the iteration points executed earlier i. e. $x_1 \leqslant x_2 \leqslant \dots \leqslant x_k$.
2. To compute a quantity $R(i)$ called a *characteristic* of the interval for each interval (x_{i-1}, x_i), $1 \leqslant i \leqslant k$.
3. To determine the interval (x_{t-1}, x_t) which the maximum characteristic $R(t) = \max\{R(i) : 1 \leqslant i \leqslant k\}$ corresponds to.
4. To compute the next iteration point x^{k+1} in the interval with the maximum characteristic i. e. $x^{k+1} \in (x_{t-1}, x_t)$.

A detailed description of various variants of this algorithm is presented in [5, 18, 19, 20].

One of the methods of the parallelization of the characteristic algorithms is based on the following consideration. The characteristic of an interval $R(t)$ can be considered as a measure of importance of this interval for the following search for the solution in this one. Therefore, after the selection of the trial point for the first processor in full accordance with the sequential algorithm. It is reasonable to choose the trial point for the second processor from the next interval in importance (i. e. from the interval with the next value of characteristic in the decreasing order), etc. A parallel algorithm constructed in such a way would differ from its sequential prototype by rules 3 and 4 only, namely:

3. To arrange the characteristics $R(i)$, $1 \leqslant i \leqslant \tau$ in the decreasing order, i. e. $R(t_1) \geqslant R(t_2) \geqslant \dots \geqslant R(t_{k-1}) \geqslant R(t_k)$
4. To select p highest characteristics with the indices t_j, $1 \leqslant j \leqslant p$ and to execute the trials in the points

$$x^{k+j} \in (x_{t_j - 1}, x_{t_j}), 1 \leqslant j \leqslant p$$

in the intervals corresponding to these characteristics.

Various variants of the parallel algorithms based on this idea are presented in [7, 8, 9, 10, 11, 12].

MIXED LOCAL-GLOBAL ALGORITHM

A scheme aimed at the acceleration of the search process is outlined in this subsection. The idea of acceleration is to magnify the characteristics of the intervals containing the best current estimates by introducing some factor depending on the function values estimated at the end points of the corresponding interval. Following this idea, the local characteristics of the intervals can be introduced [5, 21].

Next, it is possible to build a scheme mixing the 'global' and the 'local' decision rules, i. e. switching between these ones in some systematic way. In our experiments, we use global-to-local ratio q, specifying the number of global trials preceding each local trial.

The mixed strategy has the following features:

- both decision rules are based on the same information, so that each decision action (no matter local or global) uses the results of all the trials performed.
- non-stop global search assures the global convergence; the aim of the local refinement is to accelerate the attainment of low function values.

NUMERICAL EXPERIMENTS

Computing experiments were carried out on Lobachevsky supercomputer. Each supercomputer node included two Intel Sandy Bridge E5-2660 2.2 GHz CPU and 64 Gb RAM. Each CPU had 8 cores (total 16 cores per a node). For the implementation of the parallel algorithm, Intel C++ Compiler 14.0.2 was used.

The problems generated by GKLS generator [22] were used as the test ones. This generator allows generating the multiextremal optimization problems with the properties known in advance: number of local minima, size of attraction domains, global minimum point, function values, etc. The experiment included solving a series of 200 problems with $N = 4$ and $N = 5$. The global minimum y^* was considered to be found, if the algorithm generates a trial point y^k in its δ-vicinity $\|y^k - y^*\| \leqslant \delta$. The size of the vicinity was selected as $\delta = 0.01\|b - a\|$ where a and b were the borders of the search domain D. The maximum allowable number of trials was $K_{max} = 10^6$.

First, let us consider the results of experiments corresponding to the sequential algorithm. Let us compare initial GSA and mixed local-global algorithm. The CPU time consumption for the solving of four series of problems from Simple and Hard classes are presented in Table 1(a) and the numbers of the executed search trials subject to global-to-local ratio q are presented in Table 1(b). The row denoted as "—" corresponds to the algorithm without the local refining. The results demonstrate the reduction of the computation time as well as of the number of iterations when using the local-global search strategy. Also, it is seen that the best results have been achieved at the mixing parameters $q = 2$ and $q = 4$. The less and greater values of this parameter were insufficient. Therefore, the results of investigation of the mixed parallel algorithm will be presented for the value of parameter $q = 4$.

TABLE 1.

(a) CPU time consumption

q	$N = 4$		$N = 5$	
	Simple	Hard	Simple	Hard
—	0.08253	0.24744	0.11036	1.85801
1	0.0875	0.18375	0.08196	0.90628
2	0.07698	0.15921	0.07491	0.82082
4	0.07515	0.14893	0.07596	0.93878
8	0.07737	0.14959	0.07537	1.06467
16	0.07419	0.17232	0.08508	1.51871

(b) Number of iterations

q	$N = 4$		$N = 5$	
	Simple	Hard	Simple	Hard
—	12558	37410	15226	248275
1	11370	25956	9959	120978
2	9842	22456	9047	109995
4	9574	20942	9018	121934
8	9936	20879	8940	136174
16	9692	23992	10410	179490

The computation times for the mixed parallel algorithm subject to the number of processors employed p are presented in Table 2(a), and the peedup of the parallel algorithm relative to the sequential one is shown in Table 2(b).

The results of the experiments have shown the mixed local-global algorithm combined with the multilevel scheme of dimension reduction to demonstrate a good speedup, the best speedup has been achieved when solving the complex problems of 5-Hard class. Therefore, the proposed algorithm can be applied for the parallel solving of the multidimensional global optimization problems using the state-of-art supercomputer systems.

TABLE 2.

(a) CPU time consumption

p	$N = 4$		$N = 5$	
	Simple	Hard	Simple	Hard
1	0.083	0.247	0.110	1.858
1+1	0.043	0.084	0.159	0.575
2+1	0.039	0.058	0.091	0.316
4+1	0.058	0.039	0.055	0.171
8+1	0.031	0.036	0.048	0.103
16+1	0.061	0.061	0.071	0.129

(b) Speedup on CPU

p	$N = 4$		$N = 5$	
	Simple	Hard	Simple	Hard
2	1.91	2.94	0.70	3.23
3	2.12	4.27	1.22	5.88
5	1.42	6.28	2.01	10.89
6	2.66	6.88	2.28	18.05
17	1.34	4.03	1.54	14.40

ACKNOWLEDGMENTS

This research was supported by the Russian Science Foundation, project No 16-11-10150 Novel efficient methods and software tools for the time consuming decision making problems with using supercomputers of superior performance.

REFERENCES

[1] Famularo D., Pugliese P., Sergeyev Ya.D., Automatica **35**, 1605–1611 (1999).

[2] Barkalov, K., Polovinkin, A., Meyerov, I., Sidorov, S., Zolotykh, N., Lecture Notes in Computer Science **7979**, 154–166 (2013).

[3] Gergel, V.P., Kuzmin, M.I., Solovyov, N.A., Grishagin, V.A., International Review of Automatic Control **8(1)**, 51–55 (2015).

[4] Kvasov D.E., Sergeyev Y.D., Advances in Engineering Software **80**, 58–66 (2015).

[5] Strongin R.G., Sergeyev Ya.D., *Global optimization with non-convex constraints. Sequential and parallel algorithms* (Kluwer Academic Publishers, Dordrecht, 2000).

[6] Y. D. Sergeyev, R. G. Strongin, and D. Lera, *Introduction to global optimization exploiting space-filling curves* (Springer, 2013).

[7] Sergeyev, Y.D., Grishagin, V.A., Optimization Methods and Software **3**, 111–124 (1994).

[8] Grishagin V.A., Sergeyev Ya.D., Strongin R.G., Journal of Global Optimization **10**, 185–206 (1997).

[9] Gergel, V.P., Strongin, R.G., Lecture Notes in Computer Science **2763**, 76–88 (2003).

[10] Gergel, V.P., Strongin, R.G., Future Generation Computer Systems **21(5)**, 673–678 (2005).

[11] Barkalov, K., Ryabov, V., Sidorov, S., Lecture Notes in Computer Science **6083**, 232–240 (2010).

[12] Barkalov, K., Gergel, V., Lebedev, I., Lecture Notes in Computer Science **9251**, 307–318 (2015).

[13] Barkalov, K., Gergel, V., “Multilevel scheme of dimensionality reduction for parallel global search algorithms,” in *An International Conference on Engineering and Applied Sciences Optimization*, OPT-i (Kos Island, Greece, 46 June 2014), pp. 2111–2124.

[14] Gergel, V., Grishagin, V., Israfilov, R., Procedia Computer Science **51**, 865–874 (2015).

[15] Sergeyev Y.D., Kvasov D.E., Communications in Nonlinear Science and Numerical Simulation **21(1-3)**, 99–111 (2015).

[16] Kvasov D.E., 4OR - A Quarterly Journal of Operations Research **6(4)**, 403–406 (2008).

[17] Paulavicius R., Sergeyev Ya.D., Kvasov D.E., Zilinskas J., Journal of Global Optimization **59(2-3)**, 545–567 (2014).

[18] Gergel, V.P., Computational Mathematics and Mathematical Physics **36(6)**, 729–742 (1996).

[19] Sergeyev Ya.D., Famularo D., Pugliese P., Journal of Global Optimization **21(3)**, 317–341 (2001).

[20] Barkalov, K.A., Strongin, R.G., Computational Mathematics and Mathematical Physics **42(9)**, 1289–1300 (2002).

[21] Markin, D.L., Strongin, R.G., Computational Mathematics and Mathematical Physics **27(1)**, 33–39 (1987).

[22] Gaviano, M., Kvasov, D.E., Lera, D., Sergeev, Ya.D., ACM Transactions on Mathematical Software **29(4)**, 469–480 (2003).