Comparative Analysis of Parallel Computational Schemes for Solving Time-Consuming Decision-Making Problems

Victor Gergel(✉) and Evgeny Kozinov

Lobachevsky State University of Nizhni Novgorod, Nizhni Novgorod, Russia
gergel@unn.ru, evgeny.kozinov@itmm.unn.ru

Abstract. We consider in this paper an efficient approach to the parallel solution of complex multicriterial optimization problems using heterogeneous computing systems. The complexity of these problems can be very high since the criteria that are to be optimized can be multiextremal and the computation of criteria values can be time-consuming. In the framework of the proposed approach, the multicriterial optimization problem is reduced to the solution of a series of global optimization problems by means of the convolution of the partial criteria with different sets of parameters. To solve the series of global optimization problems, we apply an efficient information-statistical method of global search. Parallel computations are implemented through the simultaneous solution of several global optimization problems. We present in this paper a comparative analysis of various methods for parallel computations and the results of numerical experiments.

Keywords: Decision making · Multicriteria global optimization · Parallel calculations · Dimensionality reduction · Search of information · Numerical experiments

1 Introduction

Decision making is inherent to almost all fields of human activity. A variety of mathematical formulations have been proposed for the formal description of decision-making problems. These problems relate to many classes of optimization approaches, such as unconstrained optimization, nonlinear programming, global optimization, etc. Multicriterial optimization (MCO) problem statements are widely used in the most complex situations of decision making. An opportunity to set several criteria is a distinctive property of MCO problems which allows for a more precise formulation of the requirements to the optimality of the chosen decisions. MCO is currently a field of intensive studies; see, for example, the monographs [1–6] and reviews of scientific and practical results in [7–10].

The possibility of contradictions between the partial criteria of efficiency, which makes it impossible to achieve the optimal (the best) values for all partial

L. Sokolinsky and M. Zymbler (Eds.): PCT 2019, CCIS 1063, pp. 107–121, 2019.
https://doi.org/10.1007/978-3-030-28163-2_8

criteria simultaneously, is an important feature of multicriterial optimization problems. As a result, the determination of some compromised (efficient, nondominated) decisions, in which the achieved values with respect to some partial criteria satisfy the given requirements for the necessary level of efficiency, is usually taken as solution of an MCO problem.

The present work is devoted to solving MCO problems which are used for the description of decision-making problems in the design of complex objects and systems. In such applications, the partial criteria can take a complex multiextremal form, and therefore computing the values of the criteria and constraints may require a large number of computations.

Besides, an opportunity for correcting the MCO problem statements when changing the perceptions of the requirements to the optimality of the chosen decisions is allowed within the framework of the considered approach. Thus, in the case of redundancy of the partial criteria set, an opportunity to transform some criteria into constraints is allowed. Or, otherwise, if the set of feasible variants is insufficient, some constraints can be transformed into criteria, and so on. Such an opportunity to change the MCO problem statements is a source of additional computational costs of the search for optimal decisions.

In the present paper, we present the results of the investigations we have performed on generalization of decision-making problem statements [11,12] and on the development of highly efficient global optimization methods that use all the search information obtained in the course of computations [13–17]. Parallel algorithms developed earlier are presented in [14–17]. The proposed algorithms are able to efficiently use hundreds and thousands of computing cores [18]. Within the scope of this paper, a comparative analysis of the implemented parallel algorithms on modern computing systems is carried out. We also compare implementations of algorithms for shared memory, distributed memory, and heterogeneous computing systems. As a result of a comparative analysis, an optimal running configuration is selected.

The structure of the paper is as follows. In Sect. 2, we formulate the multicriterial optimization problem. In Sect. 3, we consider the reduction of multicriterial problems to scalar optimization ones by means of minimax convolution of the partial criteria and dimensionality reduction through the use of Peano space-filling curves. In Sect. 4, we provide arguments to substantiate the possibility of enhancing the efficiency of computations by reusing search information. In Sect. 5, we describe the general organization scheme of parallel computations allowing the maximum use of the computing potential of modern supercomputer systems. In Sect. 6, we present the results of numerical experiments confirming the viability of the proposed approach. Finally, in the Conclusions, we discuss the results obtained and suggest possible directions for future research.

2 Statement of the Multiple Multicriterial Optimization Problem

We propose the following generalized multilevel model for the formal description of the process of search for efficient variants in complex decision-making problems.

1. At the highest level, a decision-making problem (DMP) is defined within the framework of the proposed model. The optimal values of the parameters should be determined for this problem according to the available requirements to optimality. In the most general form, a DMP can be defined by a vector function of characteristics,

$$w(y) = \big(w_1(y), w_2(y), \dots, w_M(y)\big),\ y \in D, \tag{1}$$

where $y = (y_1, y_2, \dots, y_N)$ is the vector of constructive parameters and $D \in R^N$ is the domain of feasible values, which usually is an N-dimensional hyperinterval,

$$D = \{y \in R^N : a_i \le y_i \le b_i,\ 1 \le i \le N\}, \tag{2}$$

for two given vectors a and b.

The values of the characteristics $w(y)$ are supposed to be nonnegative. Moreover, when these values decrease, the efficiency of the chosen variants increases. It is also assumed that the characteristics $w_j(y)$, $1 \le j \le M$, can be multiextremal and computing their values can require large numbers of computations. In addition, within the framework of the approach under consideration in this paper, we assume that the characteristics $w_j(y)$, $1 \le j \le M$, satisfy the Lipschitz condition

$$|w_j(y_1) - w_j(y_2)| \le L_j \|y_1 - y_2\|,\ 1 \le j \le M, \tag{3}$$

where L_j is the Lipschitz constant corresponding to the characteristic $w_j(y)$, $1 \le j \le M$, and $\| \cdot \|$ denotes the Euclidean norm in R^N. It is important to note that the fulfillment of the Lipschitz condition corresponds to practical applications. Indeed, small variations of the parameters $y \in D$ lead, as a rule, to limited variations of the corresponding values of the characteristics $w_j(y)$, $1 \le j \le M$.

In general, the DMP model is defined by the following set of elements:

$$S = \langle y, w(y), D, a, b \rangle. \tag{4}$$

This model is supposed to remain the same and does not change throughout the course of computations.

2. The requirements to the optimality of the chosen variants $y \in D$ of the DMP can be defined in the following way. First of all, we should find characteristics $w_j(y)$, $1 \le j \le M$, for which the achievement of the minimum possible values is necessary. By defining a set of indices $F = \{i_1, i_2, \dots, i_s\}$ of such characteristics, we define the vector criterion of efficiency

$$f(y) = \big(w_{i_1}(y), w_{i_2}(y), \dots, w_{i_s}(y)\big). \tag{5}$$

Sufficient levels of efficiency for characteristics $w_j(y)$, $1 \leq j \leq M$, not included in the vector criterion of efficiency defined by a set of indices $G = \{j_1, j_2, \ldots, j_m\}$ should be defined by a tolerance vector $q = (q_1, q_2, \ldots, q_m)$. The availability of tolerances enables one to define a vector function of constraints

$$g(y) = \big(g_1(y), g_2(y), \ldots, g_m(y)\big),\ g_l(y) = w_{j_l}(y) - q_l, j_l \in G,\ 1 \leq l \leq m. \quad (6)$$

The constraints $g(y)$ define the feasible search domain

$$Q = \{y \in D : g(y) \leq 0\}. \quad (7)$$

With the criteria of efficiency and constraints formulated in this way, we can pose the multicriterial optimization (MCO) problem

$$f(y) \rightarrow \min,\ y \in Q, \quad (8)$$

defined on the basis of the model S from (4) using the elements listed above:

$$P = \langle S, F, G, q \rangle. \quad (9)$$

The scheme proposed covers many existing optimization problem statements. In the case $s = 1$, $m = 0$, the general statement becomes a global optimization problem. For $s = 1$ and $m > 0$, the general statement defines a nonlinear programming problem. When $s > 1$ and $m > 0$, it becomes a constrained multicriterial problem.

3. The use of multicriterial optimization problem statements reduces the complexity of the formal description of decision-making problems since it becomes possible to define several partial criteria instead of defining a unified "global" criterion of efficiency, as is necessary in MCO problems. No doubt, such an approach is a simpler task for the decision maker determining the requirements to the optimality of the decisions made. It is worth noting also that the multicriterial definition of the efficiency criteria corresponds to the practice of decision making in various fields of applications.

The scheme of the above-considered MCO problem statement was first proposed in [11] and was widely used in the solution of many applied decision-making problems. At the same time, the results of the practical application of this approach have demonstrated that the formulation of a single MCO problem statement may be difficult when altering the ideas of the necessary requirements to optimality. Thus, in the case of redundancy of the partial criteria set, the transformation of some criteria into constraints might be appropriate. Or, vice versa, if a small feasible domain Q is obtained, the loosening of the tolerances q or transforming some constraints into criteria may be required. As a result, an opportunity for simultaneous formulation of several MCO problems,

$$\mathbb{P}_t = \{P_1, P_2, \ldots, P_t\}, \quad (10)$$

will be allowed within the framework of the extended model of optimal choice as a further development of scheme (1)–(9). It is worth noting that the set of problems $\mathbb{P}$ can change in the course of computations as a consequence of the addition of new problems or the elimination of already existing ones:

$$\mathbb{P}' = \mathbb{P} +/- \mathbb{P}. \tag{11}$$

The solution of the problems from set $\mathbb{P}$ can be performed sequentially or simultaneously in time-sharing mode or in parallel if several computing devices are available. Undoubtedly, the simultaneous method of solving the problems is preferable since the results obtained in the course of computations allow to adjust promptly the current set of problems $\mathbb{P}$. In the case of parallel computations, the total time of problem solving can be reduced substantially.

In general, the model (1)–(11), proposed for the search of optimal decisions, defines a new class of optimization problems, namely multiple multicriterial global optimization (MMGO) problems.

3 Successive Reduction of Multicriterial Global Optimization Problems

As it has been mentioned above, the partial criteria of efficiency in MCO problems are usually contradictory. This feature of MCO problems implies that the attainment of optimal (the best) values with respect to all partial criteria simultaneously cannot be ensured. In such a situation, particular compromised (effective, non-dominated) decisions, in which the achieved values with respect to some specific partial criteria cannot be improved without simultaneous worsening with respect to some other criteria, are usually understood as solutions to an MCO problem. In the limiting case, it may be required to find all effective (Pareto-optimal) decisions $PD(Q) \subset Q$.

Due to the high relevance of this topic, a large number of methods have been developed and widely used for the solution of MCO problems (see, for example, [2–10]). Some of these algorithms ensure the attainment of the numerical estimates of the whole Pareto set $PD(Q)$ [3,27–29]. Along with the usefulness of this approach (all effective decisions of the MCO problem are found), the use of these methods appears to be difficult due to the high computational complexity of finding the estimates of the Pareto set. Besides, the estimate of the whole set $PD(Q)$ might be redundant in cases when it would suffice to find several particular effective decisions to solve the MCO problem (this happens quite often in practice). As a result, the more extensively used approach to the solution of MCO problems is based on the scalarization of the vector criterion into some general scalar criterion of efficiency, the optimization of which can be performed with the use of already existing optimization methods. Among these, we can mention the weighted sum method, the compromise programming method, the weighted min-max method, goal programming, and many other algorithms (see, for example, [2–6,22]).

The general approach used in the present work consists in the reduction of the solution of an MMGO problem to the solution of a series of single-criterion global optimization problems[1]:

$$\min \varphi(y) = F(\lambda, y),\ g(y) \le 0,\ y \in D, \tag{12}$$

where $g(y)$ is the vector function of constraints from (6); $F(\lambda, y)$ is the minimized convolution of the partial criteria $f_i(y(x))$, $1 \le i \le s$, of MCO problem (5),

$$F(\lambda, y) = \max\left(\lambda_i f_i(y), 1 \le i \le s\right), \tag{13}$$

using the vector of weighting coefficients

$$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_s) \in \Lambda \subset R^s : \sum_{i=1}^{s} \lambda_i = 1,\ \lambda_i \ge 0,\ 1 \le i \le s. \tag{14}$$

The approach we have developed includes one more step of conversion of the problems $F(\lambda, y)$ from (12) being solved, namely to perform a reduction of dimensionality using Peano space-filling curves (evolvents) $y(x)$ which provide an unambiguous mapping of the interval $[0, 1]$ onto an N-dimensional hypercube D [19,21]. As a result of this reduction, the multidimensional global optimization problem (12) is reduced to the one-dimensional problem

$$F(\lambda, y(x^*)) = \min\left\{F(\lambda, y(x)) : g(y(x)) \le 0, x \in [0, 1]\right\}. \tag{15}$$

The considered dimensionality reduction scheme superimposes the multidimensional problem with the Lipschitzian minimized function $F(\lambda, y)$ to a one-dimensional problem, in which the reduced function $F(\lambda, y(x))$ satisfies a uniform Hölder condition, i.e.

$$\left|F(\lambda, y'(x)) - F(\lambda, y''(x))\right| \le H\|x' - x''\|,\ x', x'' \in [0, 1], \tag{16}$$

where the constant H is defined by the relation $H = 2L\sqrt{N+3}$, N is the dimensionality of the optimization problem from (1), and L is the Lipschitz constant for the function $F(\lambda, y)$ from (12). The dimensionality reduction makes it possible to apply (after the necessary generalization) many well-known highly efficient one-dimensional global optimization algorithms for the solution of problems (12) (see, for instance, [19,21,30–36]).

To obtain several effective decisions (or to estimate the whole Pareto domain), problem (12) should be solved several times for the corresponding sets of values of the elements of the weighting vector $\lambda \in \Lambda$. In this case, the set of MCO problems $\mathbb{P}$ from (10) which are required for solving the initial decision-making problem is transformed into a wider set of scalar optimization problems (12)

$$\mathbb{F}_T = \{F_i(\lambda_i, y) : \lambda_i \in \Lambda, 1 \le i \le T\}, \tag{17}$$

where it is possible that several problems (12) with different values of the coefficients $\lambda \in \Lambda$ correspond to each problem $P \in \mathbb{P}$.

[1] On account of the initial assumptions on possible multiextremality of the characteristics $w_j(y)$, $1 \le j \le M$, from (1).

4 Improving the Efficiency of Computations by Reusing Search Information

As we already mentioned, the solution of MCO problems P from (10) and the corresponding global optimization problems F from (15) may require a large number of computations even when solving a single particular problem. In the case when a series of problems from the sets $\mathbb{P}$ and $\mathbb{F}$ is to be solved, the required quantity of computations can be much larger. And, therefore, overcoming the computational complexity of decision-making problems of the considered class is a necessary condition for the practical application of the proposed approach. An intensive use of the whole search information obtained in the course of computations can be a viable approach to the solution of this problem.

The numerical solution of global optimization problems is usually reduced to a sequential computation of the values of the characteristics $w(y)$ at the points y^i, $0 \leq i \leq k$, of the search domain D [19,23–26]. The data obtained as a result of the computations can be represented as a matrix of search information (MSI):

$$\Omega_k = \{(y^i, w^i = w(y^i))^T : 1 \leq i \leq k\}. \tag{18}$$

By scalarization of vector criterion (12) and application of dimensionality reduction (15), Ω_k in (18) can be transformed into the matrix of the search state (MSS),

$$A_k = \{(x_i, z_i, g_i, l_i)^T : 1 \leq i \leq k\}, \tag{19}$$

where

- x_i, $1 \leq i \leq k$, are the reduced points of the executed global search iterations at which the criteria values have been computed;
- z_i, g_i, $1 \leq i \leq k$, are the values of the scalar criterion and of the constraints at points x_i, $1 \leq i \leq k$, for the current optimization problem $F(\lambda, y(x))$ from (15) which is being solved, i.e.

$$z_i = \varphi(x_i) = F(\lambda, y(x_i)),\ g_i = g(y(x_i)),\ 1 \leq i \leq k; \tag{20}$$

- l_i, $1 \leq i \leq k$, are the indices of the global search iterations at which the points x_i, $1 \leq i \leq k$, were generated; these indices are used to store the correspondence between the reduced points and the multidimensional ones at the executed iterations, i.e.

$$y^j = y(x_i),\ j = l_i,\ 1 \leq i \leq k. \tag{21}$$

The matrix of the search state A_k contains search information transformed into the current scalar reduced problem (15) which is being solved. Moreover, the search information in the MSS is arranged according to the values of the coordinates of the points x_i, $1 \leq i \leq k$, for a more efficient execution of the global search algorithms; the arranged representation of the points is reflected by the use of a lower index:

$$x_1 \leq x_2 \leq \dots \leq x_k. \tag{22}$$

The representation of the search information obtained in the course of computations in the form of matrices Ω_k and A_k provides the basis for the development of efficient optimization search procedures. The availability of such information allows to perform an adaptive choice of the points for the execution of the global search iterations taking into account the results of all computations completed earlier,

$$y^{k+1} = Y(\Omega_k),\ k = 1, 2, \ldots \tag{23}$$

where Y is the rule for computing the points y^i, $0 \leq i \leq k$, of the applied optimization algorithm. Such an adaptive choice of the points of executed search iterations can accelerate the determination of the effective decisions. In the case of global optimization problems, the accumulation of the entire search information and the use of rules of type (23) is indeed mandatory. Any reduction of the search information in the determination of the points to be executed would result in the execution of excessive global search iterations.

It should be noted that the availability of the MSS Ω_k from (18) makes it possible to transform the results of all preceding computations z_i, $1 \leq i \leq k$, stored in the MSS A_k, into the values of the current optimization problem $F(\lambda, y(x))$ from (15). These recalculations are based on the values of the characteristics w^i, $1 \leq i \leq k$, which were computed earlier and stored in Ω_k without any repeated time-consuming computation of the values of $w(y)$ from (1). The updated values z_i, $1 \leq i \leq k$, can be used both for any new parameters of the MCO problem statement P from (9) and for any new values of the convolution coefficients λ from (14), i.e.

$$w_i \xrightarrow{\lambda, P} (z_i, g_i),\ 1 \leq i \leq k,\ \forall \lambda \in \Lambda,\ P \in \mathbb{P}. \tag{24}$$

Therefore, all the search information can be used to continue the computations in full. An efficient use of this information becomes an important requirement in the selection of the methods used to solve the problems $F(\lambda, y(x))$ from (15). The reuse of search information can provide a continuous decrease in the number of computations required to solve each subsequent optimization problem, up to the execution of just a few iterations only to find the next effective decision.

5 Parallel Computations in Multiple Multicriterial Global Optimization Problems

The sequential solution of the set of problems $\mathbb{F}$ from (17) to find several globally optimal decisions increases substantially the number of computations required to solve the decision-making problems. The parallel solution of the problems $F(\lambda, y(x))$ from the set $\mathbb{F}$ is a promising and simple method of accelerating the computations. For sufficiently large numbers of computer nodes (processors), the time required for the solution of the whole set of problems $\mathbb{F}$ is determined by the computation time for the problem $F(\lambda, y(x))$, whose solution takes the longest time. This approach can be implemented quite easily. However, it does not solve the problem of computational complexity when it is necessary to expand the set of problems $\mathbb{F}$ throughout the course of the computations.

The method considered above for the organization of parallel computations can be further developed by using the information compatibility of the problems of set $\mathbb{F}$. Indeed, as we noted in Sect. 4, the values of the characteristics w^i, $1 \le i \le k$, computed when solving any problem $F_1(\lambda, y(x))$, can be transformed into the values of any other problem $F_2(\lambda, y(x))$. This result provides the basis for the following general scheme of parallel computations for the simultaneous solution of the problems of set $\mathbb{F}$.

1. The Distribution of the Problems. Before starting the computations, the problems of the set $\mathbb{F}$ from (17) must be distributed among the computing nodes of a multiprocessor system. This distribution can be quite varied: for solving one problem from the set $\mathbb{F}$, various numbers of computing cores $(q, q \ge 1)$ and computing nodes $(p, p \ge 1)$ can be allocated. An opportunity to use several computing nodes $p > 1$ for the solution of the same problem is provided by applying multiple evolvents $y_i(x)$, $1 \le i \le p$ (see [13,14]). Provided that we are not planning to employ the processor cores of a computing node in full when solving one problem from the set $\mathbb{F}$, we could use the same node to solve several optimization problems, depending on the number of processors available at the computer node and the number of cores in each of them.

2. The Choice of the Optimization Algorithms. The proposed scheme of parallel computations is a general one: various methods of multiextremal optimization can be applied for solving the problems on each computing node (see, for example, [19,21,30–36]). The main condition imposed upon the selection of the algorithms is that the methods must use the search information Ω_k and A_k to increase the efficiency of global search. As it was already noted above, in Sect. 3, the global optimization algorithms should be generalized to solve the reduced one-dimensional global optimization problems of type $F(\lambda, y(x))$ from set $\mathbb{F}$. The possibility to apply the proposed approach was substantiated using as example efficient global search algorithms developed within the framework of the information-statistical theory of multiextremal optimization (see [12–21]).

3. The Execution of Global Search Iterations. The execution of global search iterations for each problem $F(\lambda, y(x))$ from the set $\mathbb{F}$ is performed in parallel. The execution of every iteration on each computing node includes the following operations:

1. The choice of q, $q \ge 1$, points x^i, $1 \le i \le q$, of the next global search iteration is performed according to the optimization method rule given in (23). The points are chosen taking into account the available search information Ω_k and A_k. The number q, $q \ge 1$, of generated points is determined by the number of computing cores employed to solve the problem $F(\lambda, y(x))$.
2. For each chosen one-dimensional point $x^i \in [0, 1]$, $1 \le i \le q$, of the current global search iteration, the multidimensional image $y^i \in D$, $1 \le i \le q$, is determined by the mapping $y(x)$. Then, each computed image $y^i \in D$, $1 \le$

$i \leq q$, is sent to all employed computing nodes, so as to prevent that the same points of the domain D are chosen more than once when solving the problems of the set $\mathbb{F}$.

3. The values of the characteristics $w(y)$ from (1) are computed at all points $y^i \in D$, $1 \leq i \leq q$. For each point $y^i \in D$, $1 \leq i \leq q$, these computations are performed in parallel using different computing cores. The computed values of the characteristics $w(y)$ are sent to all employed computing nodes to include the data obtained into the search information Ω_k and A_k.

4. Updating Search Information. Before starting the next global search iteration, the availability of data sent from other computing units (processors or cores) is checked; the received data should be included in the search information.

According to the suggested computational scheme of parallel computations, each computing unit contains an identical copy of the matrix of search information Ω_k from (18); the matrices A_k from (19) contain different sets of global search points x_i, $1 \leq i \leq k$, because of the use of different evolvents $y(x)$ for the dimensionality reduction and the different values of the scalar criterion and the constraints z_i, g_i, $1 \leq i \leq k$, corresponding to the problems $F(\lambda, y(x))$ from the set $\mathbb{F}$.

Within the framework of such computational scheme, it becomes possible to expand the set of problems $\mathbb{F}$ being solved at any moment of the computations. Indeed, to solve a new problem $F(\lambda, y(x))$, it is enough to allocate an additional computing unit, copy the set Ω_k, and create the corresponding matrix A_k.

It is clear that the increase in global search efficiency due to the use of the sets of search information Ω_k and A_k depends on the optimization algorithms employed. An analysis of the elements of the above-considered scheme of parallel computations applied to information-statistical global optimization algorithms can be found in [12–17]. In Sect. 6, we present the results of the numerical experiments carried out for a complete evaluation of the efficiency of the scheme we used to choose the best parameters for using the computing resources of high-performance supercomputer systems.

6 Results of Numerical Experiments

The numerical experiments were carried out on the "Lobachevsky" supercomputer, at the University of Nizhni Novgorod (operation system: CentOS 6.4, management system: SLURM). Each supercomputer node is equipped with two Intel Sandy Bridge E5-2660 processors (2.2 GHz, 64 Gb RAM). Each processor has eight cores (making a total of 16 CPU cores available on each computer node). We used the Intel C++ 17.0 compiler to generate the executable program code.

The numerical experiments, conducted according to the general scheme of parallel computations considered in Sect. 5, made use of efficient algorithms of multicriterial multiextremal optimization [12–17] developed within the framework of the information-statistical theory of global optimization [19]. The efficiency of these algorithms has been proven by many numerical experiments.

In the present work, we present the results of a series of numerical experiments performed to choose the best parameters of parallel computations (the number of computing cores and computer nodes employed, the number of multicriterial optimization subproblems to be solved simultaneously, the number of evolvents used for the dimensionality reduction, etc.). Each experiment included the solution of 30 MCO problems, each one six-dimensional and five-criterial, i.e. $N = 6$, $s = 5$. Multiextremal functions defined by the GKLS generator (see [37]) were used as criteria. This generator produces multiextremal optimization problems having properties given a priori, such as the number of local minima, sizes of their attractors, the global minimum point, the value of the objective function at this point, etc.

The values of the parameters used in the numerical experiments were as follows. The computation stop condition in the solution of the MCO problems was set for a predefined accuracy $\varepsilon = 0.05$. The estimates of the Pareto domain $PDA(f, D)$ were constructed by solving 100 subproblems of the family $\mathbb{F}$ from (17) for various values of the coefficients $\lambda \in \Lambda$. In the experiments, 10 computing nodes of the supercomputer were used, each having two 8-core processors (which gives a total of 160 computing cores used for computations in each experiment).

At the beginning of the series of experiments, the parallel computations were performed using only a single supercomputer node (2 processors, 16 computing cores using shared memory). The results of these experiments are contained in Table 1.

Table 1. Evaluation of the efficiency of a single supercomputer node for the solution of 30 six-dimensional five-criterial MCO problems

Cores	Search information	Iterations	S1	S2
1	Not used	26 813 722.7	1.0	
1	Used	9 103 069.6	2.9	1.0
8	Used	1 291 720.0	20.8	7.0
16	Used	609 169.5	44.0	14.9

The number of computing cores employed in the experiments is given in the *Cores* column. The *Search information* column contains information on reuse of the search information when solving the series of subproblems of the family $\mathbb{F}$ from (17) for different values of the coefficients $\lambda \in \Lambda$. The *Iterations* column presents the average number of global search iterations executed in the solution of a single subproblem of the MCO problem. The *S1* column shows the overall speedup achieved in parallel computations for the corresponding number of computing cores. Finally, the *S2* column shows the speedup of computations with respect to the serial optimization algorithm, in which the search information is used.

As follows from the results in Table 1, even when using a single computing core, one can achieve a computation speedup by almost a factor of three by reusing the search information obtained during the optimization. The overall speedup achieved in parallel computations is 44 times using the 16 cores of a single supercomputing node.

Ten supercomputer nodes (i.e. a total of 160 computing cores) were used in all the experiments. The parameters of the experiments were the following: the number of subproblems of the family $\mathbb{F}$ from (17) being solved simultaneously, the number of computing nodes employed for solving a single MCO subproblem, and the number of test problems (produced by the GKLS generator) that are solved simultaneously (a total of 30 six-dimensional five-criterial test MCO problems were solved in each experiment). The results of the numerical experiments are summarized in Table 2.

Table 2. Evaluation of efficiency of ten supercomputing nodes for the solution of a series of 30 six-dimensional five-criterial MCO problems

Nodes	NumMCO	SubProblems	SubNodes	Iterations	S1	Scaling
1	1	1	1	609 169.5	44.0	1.0
10	1	10	1	75 045.8	357.3	8.1
10	1	5	2	77 643.6	345.3	7.8
10	1	1	10	108 700.0	246.7	5.6
10	2	5	1	76 819.0	349.1	7.9
10	2	1	5	81 132.8	330.5	7.5
10	**5**	**2**	**1**	**66 618.6**	**402.5**	**9.1**
10	5	1	2	77 995.0	343.8	7.8
10	10	1	1	73 999.7	362.3	8.2

The number of employed supercomputing nodes is given in the *Nodes* column. The *NumMCO* column contains the number of test MCO problems (produced by the GKLS generator) solved simultaneously. The *SubProblems* column lists the number of subproblems of the family $\mathbb{F}$ from (17) solved simultaneously. The *SubNodes* column gives the number of computing nodes employed for solving one MCO subproblem. The *Iterations* column indicates the average number of global search iterations executed for the solution of one single subproblem of the MCO problem. The *S1* column indicates the overall speedup achieved in parallel computations for the corresponding number of supercomputing nodes. Finally, the *Scaling* column shows the speedup of parallel computations with respect to the results obtained when using a single computing node.

As follows from the results presented in Table 2, the best speed up in the considered series of experiments was achieved for the simultaneous solution of five MCO problems of the test class when two subproblems of the family $\mathbb{F}$ from (17) were solved in parallel for each problem, using a single supercomputing node for each subproblem (the parameters of the corresponding experiments are shown in bold in Table 2). In this case, the speedup factor achieved in parallel computations exceeded 400 (it should be noted that we used in the experiment 10 computing nodes with a total of 160 cores).

7 Conclusions

In the present paper, we proposed an efficient computational scheme of parallel computations for solving complex multicriterial optimization problems with non-convex constraints in which the optimization criteria can be multiextremal and the computation of criteria values can require a large quantity of computations. The approach proposed is based on the reduction of the multicriterial problems to nonlinear programming problems by means of minimax convolution of the partial criteria, dimensionality reduction through the use of Peano space-filling curves, and application of efficient information statistical and global optimization methods which implement a novel index scheme of accounting for the constraints instead of the penalty functions, which are commonly used.

The developed general computational scheme includes concurrent parallel computational methods for computing systems with shared memory, distributed parallel computations using multiple mappings for dimensionality reduction, and multilevel nesting of parallel computations for high-performance computing systems. In general, the proposed computational scheme can provide for an efficient application of high-performance computing systems with large numbers of cores/processors for the solution of complex global optimization problems. Besides, this general scheme can be used for the organization of parallel computations for a wide range of algorithms to solve time-consuming problems of decision making (in particular, for information-statistical global optimization algorithms).

We carried out a series of experiments to determine the optimal parameters of parallel computations. The results of our research showed that the suggested approach leads to a significant reduction of the computational costs of the solution of multicriterial optimization problems with non-convex constraints.

Finally, this research has shown the viability of the approach we have developed, but it requires further investigation. First of all, numerical experiments on the solution of multicriterial optimization problems should be conducted for larger numbers of partial criteria of efficiency and for larger dimensions of the optimization problems.

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

References

1. Parnell, G.S., Driscoll, P.J., Henderson, D.L. (eds.): Decision Making in Systems Engineering and Management, 2nd edn. Wiley, Hoboken (2011)
2. Miettinen, K.: Nonlinear Multiobjective Optimization. Springer, Heidelberg (1999)
3. Marler, R.T., Arora, J.S.: Multi-Objective Optimization: Concepts and Methods for Engineering. VDM Verlag (2009)
4. Ehrgott, M.: Multicriteria Optimization, 2nd edn. Springer, Heidelberg (2010)
5. Collette, Y., Siarry, P.: Multiobjective Optimization: Principles and Case Studies (Decision Engineering). Springer, Heidelberg (2011)
6. Pardalos, P.M., Žilinskas, A., Žilinskas, J.: Non-Convex Multi-Objective Optimization. Springer, Heidelberg (2017)
7. Figueira, J., Greco, S., Ehrgott, M. (eds.): Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, New York (2005)
8. Hillermeier, C., Jahn, J.: Multiobjective optimization: survey of methods and industrial applications. Surv. Math. Ind. **11**, 1–42 (2005)
9. Zavadskas, E.K., Turskis, Z., Kildiene, S.: State of art surveys of overviews on MCDM/MADM methods. Technol. Econ. Dev. Econ. **20**, 165–179 (2014)
10. Cho, J.-H., Wang, Y., Chen, I.-R., Chan, K.S., Swami, A.: A survey on modeling and optimizing multi-objective systems. IEEE Commun. Surv. Tutor. **19**(3), 1867–1901 (2017)
11. Strongin, R.G., Gergel, V.P., Markin, D.L.: Multicriterion multiextreme optimization with nonlinear constraints. In: Lewandovski, A., Volkovich, V. (eds.) LNEMS, vol. 351, pp. 120–127. Springer, Laxenburg (1988)
12. Gergel, V.P., Kozinov, E.A.: Accelerating multicriterial optimization by the intensive exploitation of accumulated search data. AIP Conf. Proc. **1776**, 090003 (2016). https://doi.org/10.1063/1.4965367
13. Gergel, V.P., Kozinov, E.A.: Efficient multicriterial optimization based on intensive reuse of search information. J. Glob. Optim. **71**(1), 73–90 (2018). https://doi.org/10.1007/s10898-018-0624-3
14. Gergel, V.P., Kozinov, E.A.: Accelerating parallel multicriterial optimization methods based on intensive using of search information. Procedia Comput. Sci. **108**, 1463–1472 (2017)
15. Gergel, V., Kozinov, E.: Parallel computing for time-consuming multicriterial optimization problems. In: Malyshkin, V. (ed.) PaCT 2017. LNCS, vol. 10421, pp. 446–458. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62932-2_43
16. Gergel, V., Kozinov, E.: Efficient methods of multicriterial optimization based on the intensive use of search information. In: Kalyagin, V., Nikolaev, A., Pardalos, P., Prokopyev, O. (eds.) Models, Algorithms, and Technologies for Network Analysis. PROMS, vol. 197, pp. 27–45. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56829-4_3
17. Gergel, V., Kozinov, E.: An approach for parallel solving the multicriterial optimization problems with non-convex constraints. In: Voevodin, V., Sobolev, S. (eds.) RuSCDays 2017. CCIS, vol. 793, pp. 121–135. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71255-0_10
18. Gergel, V., Kozinov, E.: GPU-based parallel computations in multicriterial optimization. In: Voevodin, V., Sobolev, S. (eds.) RuSCDays 2018. CCIS, vol. 965, pp. 88–100. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05807-4_8
19. Strongin, R., Sergeyev, Y.: Global Optimization with Non-convex Constraints. Sequential and Parallel Algorithms. Kluwer Academic Publishers, Dordrecht (2nd ed. 2013, 3rd ed. 2014)

20. Strongin, R., Gergel, V., Grishagin, V., Barkalov, K.: Parallel Computations for Global Optimization Problems. Moscow State University Press, Moscow (2013). (in Russian)
21. Sergeyev, Y.D., Strongin, R.G., Lera, D.: Introduction to Global Optimization Exploiting Space-Filling Curves. Springer, Heidelberg (2013)
22. Eichfelder, G.: Scalarizations for adaptively solving multi-objective optimization problems. Comput. Optim. Appl. **44**, 249–273 (2009)
23. Floudas, C.A., Pardalos, M.P.: Recent Advances in Global Optimization. Princeton University Press, Princeton (2016)
24. Locatelli, M., Schoen, F.: Global Optimization: Theory, Algorithms, and Applications. SIAM (2013)
25. Arora, R.K.: Optimization: Algorithms and Applications. CRC Press, Boca Raton (2015)
26. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms. Wiley, Hoboken (2006)
27. Evtushenko, Y.G., Posypkin, M.A.: A deterministic algorithm for global multi-objective optimization. Optim. Methods Softw. **29**(5), 1005–1019 (2014)
28. Žilinskas, A., Žilinskas, J.: Adaptation of a one-step worst-case optimal univariate algorithm of bi-objective Lipschitz optimization to multidimensional problems. Commun. Nonlinear Sci. Numer. Simul. **21**, 89–98 (2015)
29. Markin, D.L., Strongin, R.G.: Uniform estimates for the set of weakly effective points in multi-extremum multicriterion optimization problems. Comput. Math. Math. Phys. **33**(2), 171–179 (1993)
30. Piyavskij, S.A.: An algorithm for finding the absolute extremum of a function. Comput. Math. Math. Phys. **12**, 57–67 (1972)
31. Shubert, B.O.: A sequential method seeking the global maximum of a function. SIAM J. Numer. Anal. **9**, 379–388 (1972)
32. Breiman, L., Cutler, A.: A deterministic algorithm for global optimization. Math. Program. **58**(1–3), 179–199 (1993)
33. Törn, A., Žilinskas, A.: Global Optimization. LNCS, vol. 350. Springer, Heidelberg (1989). https://doi.org/10.1007/3-540-50871-6
34. Evtushenko, Y.G.: Numerical Optimization Techniques. Translations Series in Mathematics and Engineering. Springer, Berlin (1985)
35. Sergeyev, Y.D.: Global one-dimensional optimization using smooth auxiliary functions. Math. Program. **81**, 127–146 (1998)
36. Gergel, V.P.: A method of using derivatives in the minimization of multiextremum functions. Comput. Math. Math. Phys. **36**, 729–742 (1996)
37. Gaviano, M., Kvasov, D.E., Lera, D., Sergeyev, Y.D.: Software for generation of classes of test functions with known local and global minima for global optimization. ACM Trans. Math. Softw. **29**(4), 469–480 (2003)