

Accelerating Multicriterial Optimization by the Intensive Exploitation of Accumulated Search Data

Victor Gergel^{1,a)} and Evgeny Kozinov^{1,b)}

¹*Lobachevsky State University of Nizhni Novgorod, Nizhni Novgorod, Russia*

^{a)}gergel@unn.ru

^{b)}evgeny.kozinov@itmm.unn.ru

Abstract. The work proposed an efficient method for solving computationally difficult multicriterial optimization problems, which are widely used to model complex optimal decision making problems. Under the suggested approach, it is assumed that partial criteria can be multi-extremal and computationally intense, and finding a solution to multicriterial problems can require the sequential computation of several efficient (Pareto-optimal) alternatives. This multiple repetition of alternative searches leads to a substantial increase in computational costs, and the problem can be overcome by means of full usage of all search information obtained during the computations. The article provides a description of the developed approach, the efficiency of which has been substantiated by the results of computational experiments.

EXPANDED STATEMENT OF MULTICRITERIAL OPTIMIZATION PROBLEMS

Multicriterial optimization (MCO) problems are used, as a rule, to model the decision making problems see, for example, [1-5]. The opportunity to define several partial criteria complies with the decision-making practice in many areas of application. At the same time, the problem definition may need to be updated during the computations, so as to expand or reduce the number of criteria.

Taking into account the above, the following expanded statement of MCO problems is suggested to ensure a more adequate description of decision making problems.

1. Before proceeding with the selection problem, it is necessary to define the object of optimization as the following set of components:

$$O = \langle y, D, w(y) \rangle, \quad (1)$$

where:

- y is the parameters vector:

$$y = (y_1, y_2, \dots, y_N)$$

(N is the dimension of the problem);

- D is the hyper-interval of possible values of the vector y , determined by the predefined vectors a and b :

$$D = \{y \in R^N : a_i \leq y_i \leq b_i, 1 \leq i \leq N\};$$

- $w(y)$ is the vector function of characteristics determining the object efficiently:

$$w(y) = (w_1(y), w_2(y), \dots, w_M(y))$$

(the characteristics can include, for example, weight, speed, cost, etc.).

It is expected that the characteristics $w_j(y)$, $1 \leq j \leq M$, can be multiextremal and require intensive computations. Without reducing the generality, the characteristics are also supposed to be non-negative, and their decreasing will

result in improved object efficiency. It is also required that the model of the optimization object remains constant during the computations and does not change.

2. The requirements for the optimal alternative of the optimization object can be determined using the following set of elements:

$$Z = \langle F, G, q \rangle, \quad (2)$$

where:

- $F = \langle i_1, i_2, \dots, i_s \rangle$ is a set of indices determining the vector *criteria* of efficiency

$$f(y) = (w_{i_1}(y), w_{i_2}(y), \dots, w_{i_s}(y)),$$

as a subset of functions from the vector function of characteristics $w(y)$;

- $G = \langle j_1, j_2, \dots, j_m \rangle$ is a set of indices determining the vector function of *constraints*

$$g(y) = (w_{j_1}(y) - q_1, w_{j_2}(y) - q_2, \dots, w_{j_m}(y) - q_m),$$

and $q = (q_1, q_2, \dots, q_m)$ is the tolerance vector. The constraints $g(y)$ set the *feasible search domain*

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

The efficiency criteria and constraints defined in this way, allow the *multicriterial optimization problem* to be defined

$$f(y) \rightarrow \min, y \in Q.$$

This proposed scheme gives a chance to change the statement during computations. Unlike the object model (1), the MCO problem (2) can be changed some constraints may be modified, tolerance values may be tightened or, on the contrary, relaxed, etc. The possibility for such variations allows to set requirements for the desired decision more adequately. As a result, the decision making in the proposed approach is declared as a multi-step procedure, where the MCO problem is reviewed at each stage and subsequently solved.

METHODS FOR SOLVING MCO PROBLEMS

1. In the MCO problem (2), partial criteria are usually controversial and there are no point $y \in D$ that would provide the best (smallest) values for all criteria simultaneously. In these cases, the solution of the MCO problem considers such point $y^* \in D$, for which partial criteria cannot be improved without decreasing the efficiency for any other criteria. These unimprovable variants are called *efficient* or *Pareto optimal*. Solving an MCO problem usually requires finding one or more (in extreme cases, the entire set) of the efficient points.

One of the widely used approaches to finding efficient alternatives is the convolution of the partial criteria $f_i(y)$, $1 \leq i \leq s$, to a single scalar function $\Phi(\lambda, y)$, which allows to reduce the problem (2) to a scalar nonlinear programming problem (see [2-4], for example):

$$\min \Phi(\lambda, y), y \in Q \quad (3)$$

where $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_s)$ is the vector of coefficients used in constructing the scalar criterion.

A definite advantage to this approach is the possibility of solving scalar problems (3) using numerous efficient nonlinear programming algorithms [5-13].

To obtain scalar criteria $\Phi(\lambda, y)$, in many cases it is possible to use a minimax convolution of the partial criteria [2-4]:

$$\Phi(\lambda, y) = \max_{1 \leq i \leq s} \lambda_i f_i(y), \quad (4)$$

$$\lambda_i \geq 0, 1 \leq i \leq s, \sum_{i=1}^s \lambda_i = 1$$

A main property of minimax convolution is the necessity and sufficiency of this approach in solving the MCO problem: optimizing $F(\lambda, y)$ produces an efficient option¹ for the MCO problem, and vice versa, any efficient point for the MCO problem can be obtained as a result of minimization of $F(\lambda, y)$, given the respective values of the convolution coefficients $\lambda_i, 1 \leq i \leq s$.

2. It should be noted that solving optimization problems (3) and (4) can require massive computations. These computations can be reduced greatly if the points for calculating criteria and constraints are only densely spaced around the globally optimal alternatives sought. This nonuniform coverage can only be supported when performing adaptive computations, when the selection of new iteration points uses the entire search data (previous iteration points and the values of the function being minimized at these points) obtained during the computations. This condition substantially complicates the computational procedures for global optimization methods, as it implies a complex computational analysis of multidimensional search data.

To reduce the complexity of multidimensional optimization methods, various dimensionality reduction schemes are used. For example, Peano *curves* or *mappings* $y(x)$ are widely used (see, for example, [6-7]), which allow to map uniquely the one-dimensional interval $[0, 1]$ onto a N-dimensional search domain. As a result of this reduction, the multidimensional global optimization problem (4) is reduced to the following univariate problem:

$$\phi(\lambda, x) = \Phi(\lambda, y(x)), x \in [0, 1]. \quad (5)$$

Using Peano mappings to reduce dimensionality became the grounds for developing a large number of efficient multiextremal optimization methods see, for example, [6-12].

3. Reducing dimensionality does not allow sufficient decreasing the computations in solving MCO problems. The biggest problem is that finding optimal decisions may require repeated solution of general multiextremal problems (3)-(5) when the MCO problem (2) is changed or new values for criteria convolution coefficients are selected. This problem can be overcome only by using the whole search data accumulated during the computations.

A numerical solution to the optimal decision making problems usually consists of repeated computations of the values of characteristics $w(y)$ at points $y^i, 0 \leq i \leq k$ within the search domain D . The results of the respective computations can be represented as a matrix (*search data matrix*, SDM):

$$\Omega_k = \{(y^i, w^i = w(y^i))^T : 0 \leq i \leq k\}.$$

The size of the search data can become rather large when solving complex multidimensional problems, but any reduction in the volume of data stored, as a rule, results in excessive iterations of the global search [6-7].

As a result of reducing the decision making problem to an MCO problem (2), scalarizing vector criteria (3) or (4), and reducing dimensionality (5), a SDM is transformed into a *search state matrix* (SSM)

$$\omega_k = \{(x_i, z_i = \phi(\lambda, x^i), f^i, g^i, w^i)^T : 0 \leq i \leq k\}.$$

which meets the statement of the optimization problem currently being solved.

Providing search information as an SSM is important for substantially reducing the computational intensity of solutions to the decision making problems. Optimization methods can use SSM for the adaptive performance of another round of search iterations (using the results of previous computations). And most importantly, the presence of SSM allows to recalculate the results of previous computations to the values of another next optimization problem (5) without any intense computations of the vector function characteristics $w(y)$ from (1). Thereby, all search data can be fully reused in continuing the computations. And as the main result, reusing the search data will require increasingly fewer computations for each subsequent optimization problem, until there are only a few iterations will be needed to find next efficient variants.

RESULTS OF COMPUTATIONAL EXPERIMENTS

To evaluate the efficiency of the proposed approach, a number of computational experiments for bicriterial univariate MCO problems without constraints has been executed, i.e., $N = 1, M = 2, s = 2, m = 0$. Multiextremal functions [14] were used as criteria for the problem defined by expression:

$$\psi(x) = A_0 + \sum_{i=1}^{14} (A_i \sin(2i\pi x) + B_i \cos(2i\pi x)), 0 \leq x \leq 1,$$

¹To be more precise, minimization of $\Phi(\lambda, y)$ can lead to weak efficient points (the set of weak efficient points includes the Pareto domain).

TABLE 1. Results of experiments with and without using the search data.

Number of problems to be solved	Without using search data		Using search data		Decreasing of executed iterations (times)
	Total number of iterations	Average per job	Total number of iterations	Average per job	
1	40	40	40	40	1
5	234	46,8	69	13,8	3,4
10	566	56,6	93	9,3	6,1
15	746	49,7	104	6,9	7,2
20	1018	50,9	142	7,1	7,2
25	1304	52,2	151	6	8,6

were used as the problem criteria and $A_0, A_i, B_i, 1 \leq i \leq 14$, are pseudo-random values uniformly distributed in the range of $[-1, 1]$.

To optimize the convolution of the criteria (5) we used a global search algorithm developed in the framework of information-statistical theory of the global optimization [6-8], with a reliability parameter $r = 2$ and search accuracy $\varepsilon = 0.001$. In each experiments performed, we solved a problem (5) with several different convolution coefficients $\lambda_i, 1 \leq i \leq s$ (from 1 to 25), with and without using the search data. The experiment results are presented in Table 1.

As the experiments presented above show, the use of search data allows the computational intensity to be reduced by up to 8 times when computing multicriterial optimization problems.

ACKNOWLEDGEMENTS

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."

CONCLUSION

This work proposed an approach to increasing the efficiency of handling computationally intensive multicriterial optimization problems by reusing search data obtained in the computations. The results of computational experiments confirm the viability of the developed approach.

Further development of this approach is needed to provide theoretical estimates for the reduction in computational intensity, and to expand substantially the range of computational experiments conducted for multidimensional multicriterial optimization problems with constraints.

REFERENCES

- [1] M. Ehrgott, *Multicriteria Optimization* (Springer, 2005 (2nd ed., 2010)).
- [2] K. Miettinen, *Nonlinear Multiobjective Optimization* (Springer, 1999).
- [3] A. Mardani et al., Economic Research-Ekonomiska IstraĀivanja **28(1)**, 516–571 (2015).
- [4] Y. Evtushenko and M. Posypkin, Optimization Methods and Software **29(5)**, 1005–1019 (2014).
- [5] V. Gergel et al., Procedia Computer Science **18**, 10–19 (2013).
- [6] R. G. Strongin and Y. D. Sergeyev, *Global optimization with non-convex constraints. Sequential and parallel algorithms* (Kluwer Academic Publishers, 2000).
- [7] Y.D. Sergeyev et al., *Introduction to global optimization exploiting space-filling curves* (Springer, 2013).
- [8] R. G. Strongin and V. P. Gergel, Future Generation Computer Systems **21**, 673–678 (2005).
- [9] K. Barkalov and V. Gergel, OPT-i 2014. An International Conference on Engineering and Applied Sciences Optimization (Kos Island, Greece, 4–6 June 2014) 2111–2124 (2014).
- [10] V. Gergel et al., Procedia Computer Science **51**, 865–874 (2015).
- [11] K. Barkalov and V. Gergel, Journal of Global Optimization 1–18 (13 February 2016).
- [12] D. Lera and Y. Sergeyev, SIAM Journal on Optimization **23(1)**, 508–529 (2013).
- [13] R. Paulavicius et al., Journal on Optimization **59(2-3)**, 545–567 (2014).
- [14] J. Hill, IEE Transactions on Systems and Cybernetics **5(1)**, 2–8 (1969).