

SOFTWARE PACKAGE FOR THE NUMERICAL SOLUTION OF NONLOCAL OPTIMIZATION PROBLEMS

Pavel S. Sorokovikov¹, and Alexander Yu. Gornov¹

¹ Optimal Control Laboratory, Matrosov Institute for System Dynamics and Control Theory of
Siberian Branch of Russian Academy of Sciences,
Lermontov Street, 134, 664033, Irkutsk, Russia
e-mail: sorokovikov.p.s@gmail.com, gornov@icc.ru

Abstract

The paper discusses the developed first version of the MEOPT (“Multi-Extremal OPTimization”) software package designed to solve multiextremal problems of parametric identification of nonlinear models. The project includes the following components: software framework that provides a uniform approach to the implementation of algorithms and includes basic functionality; libraries of optimization algorithms; collection of test problems with known solutions; tool and service modules; metacomponents. The implemented libraries of optimization algorithms include multidimensional nonconvex optimization methods and algorithms for solving auxiliary problems of global univariate search. The library of test problems includes both mathematical programming problems and tasks of optimization of dynamic systems. The software package was developed using the C language and the GCC compiler; it operates under Linux, Windows, and Mac OS operating systems. To date, prototypes of the main modules of the package have been completed. The technical testing of the first version of the software was carried out.

Keywords: Software Package, Nonconvex Optimization, Parametric Identification, Optimal Control.

1 INTRODUCTION

Computational technologies for solving problems of optimization of dynamical systems (optimal control, trajectory optimization) developed in parallel with the results in theory. In the developments below, the main efforts were directed at reducing optimal control problems to mathematical programming problems and using advanced software tools for solving finite-dimensional optimization tasks.

In 1983-1984, the first CONTROL software complex was developed in Russia, designed to solve optimal control problems [1]. The use of mathematical programming methods for solving these problems was implemented in the “Dialogue Optimization System” (DIOS) [2, 3]. This system provided the opportunity to work in a dialogue mode, allowing the use of various interactive technologies for solving the problem. The development of software packages has been carried out at the Irkutsk Computing Center of the SB RAS since its inception. These software packages were designed for both linear optimal control problems [4] and nonlinear optimization problems (software MAPR – “Mathematical programming in multidimensional problems”) [5]. MAPR has been repeatedly used to solve applied problems of optimization of dynamic systems [6]. The “software package for optimal control problems” included methods based on sufficient optimality conditions and was used to solve ecological, economic, and navigation problems [7]. Computing technologies of sequential sampling were implemented in the package of programs “Complex optimization of nonlinear controlled systems” (CONCS) [8]. The first version of the OPTCON software package for solving optimal control problems was developed by A.Yu. Gornov and has been used in practical applications for several decades (see, for instance, [9, 10]).

Boeing Computer Services has created the SOCS “Sparse optimal control software” (J.T. Betts, W.P. Huffman) [11], which is successfully used to solve applied problems in the field of robot control and navigation. Another well-known software package is the MISER3 project [12] (K.L. Teo, C.J. Goh, Hong Kong Polytechnic University), which has a solid history. The DIRCOL software package [13] (O. von Stryk, Technological Darmstadt University, Germany) is based on the use of computational technology, which is also embedded in the SNOPT and NPSOL software complexes (Systems Optimization Laboratory, Stanford University) and is focused on the solution finite-dimensional optimization problems. By combining existing mathematical programming packages (CPLEX, MINOS, SNOPT, LPSOLVE, NPSOL, DAESSA, DASOLV), a computational technology for solving optimal control problems MINOPT was developed [14] (C. Schweiger., C. Floudas, Princeton University, USA). M. Gerdt from Bundeswehr University Munich developed the Optimal Control of Ordinary-Differential Equations (OC-ODE) software [15], which is a set of routines to solve optimal control tasks that apply an automatic direct discretization method for the reducing of the optimal control problem into a finite-dimensional nonlinear optimization problem. The Dynamic Optimization Toolbox with Vector Control Parametrization (DOTcvp) was proposed and implemented by T. Hirmajer, E. Balsa-Canto, and J.R. Banga (Bioprocess Engineering Group, IIM-CSIC, Spain) [16]. The Multiple Shooting CODE for Optimal Control (Muscod-II) developed in Heidelberg University [17] is the software package for the investigation of mixed-integer nonlinear optimal control problems and is part of the NEOS platform. CasADi is an open source framework for nonlinear trajectory optimization (Department of Chemical and Biological Engineering, University of Wisconsin-Madison, USA) [18]. P.E. Rutquist (Tomlab Optimization AB, Sweden) and M.M. Edvall (Tomlab Optimization Inc., USA) developed the PROPT optimal control software [19]. P. Falugi, E. Kerrigan, and E. Wyk from the Imperial College London proposed and implemented Imperial College London Optimal Control Software (ICLOCS) [20]. Another solver for the

investigation of complex trajectory optimization problems is the PSOPT developed by V.M. Becerra from the School of Systems Engineering of the University of Reading [21]. It should also be noted the following software packages for solving problems of optimization of dynamic systems (see, for example, [22]): BNDSCO, POST, OTIS, GESOP, NTG, MIDAS, GMAT, COPERNICUS, KNITRO, DIDO, RIOTS_95, DIRECT, OPTCONTROLCENTRE, IPOPT, GPOPS, and others.

We present the developed first version of the MEOPT (“Multi-Extremal OPTimization”) software package designed to solve multiextremal problems of parametric identification of nonlinear models. The developed software differs from the other aforementioned solvers by an extensive library of nonlocal optimization algorithms of various genesis, the ability to work in several modes, and other characteristics described in the paper. The article is organized in the following way: Section 2 formulates the mathematical and technological statement of trajectory optimization problem; Section 3 presents the logical structure of the developed software package; Section 4 describes implemented functional subsystems; Section 5 reports operating modes of the software package; Section 6 concludes the paper.

2 FORMULATION OF THE PROBLEM

2.1 Mathematical statement

We described the controlled process by a system of ordinary differential equations with initial conditions:

$$\dot{x} = f(x(t), u(t), t), \quad x(t_0) = x^0, \quad (1)$$

where t is the time from the interval $[t_0, t_1]$, $x(t) = (x_1(t), x_2(t), \dots, x_n(t))$ is the vector of phase coordinates, $u(t) = (u_1(t), u_2(t), \dots, u_r(t))$ is the vector of control actions. A vector function $f(x(t), u(t), t)$ is assumed to be continuously differentiable for all arguments except t . Admissible functions are piecewise-continuous control functions $u(t)$ for any time values t belonging to the set U , where

$$U = \{u(t) \in R^r : u_l \leq u(t) \leq u_g\}, \quad (2)$$

$u_l, u_g \in R^r$ are vectors of lower and upper control constraints. The optimal control problem in the standard-setting is to find an admissible control $u^*(t)$ that delivers a minimum to the terminal functional

$$I_0(u) = \varphi_0(x(t_1)) \rightarrow \min. \quad (3)$$

There are also terminal restrictions

$$I_j(u) = \varphi_j(x(t_1)) = (\leq) 0, \quad j = \overline{1, m} \quad (4)$$

and inequality type phase constraints

$$I_j(u) = g_j(x(t), u(t), t) \leq 0, \quad j = \overline{m+1, mt}. \quad (5)$$

All functions $\varphi_j(x(t_1))$, $j = \overline{0, m}$ and $g_j(x(t), u(t), t)$, $j = \overline{m+1, mt}$ are assumed to be continuously differentiable for all arguments.

For the numerical solution of optimal control problems, the discretization of the system of differential equations and approximate methods for solving the Cauchy problem are used. We divided the time change segment into $n_u - 1$ parts and constructed a uniform grid $\{t_0 + ih, i = \overline{0, n_u - 1}\} \cup t_1$, in the nodes of which the controls and trajectories are stored (n_u is the number of sampling points).

2.2 Technological statement

To numerically solve the problem of identifying nonlinear dynamic models using a software package, it is necessary to write down the mathematical statement of the problem in the programming language C. At this stage, the right-hand sides of the controlled system of differential equations, the vector of initial states, the time interval, and the set of admissible controls are set.

The technological formulation of the problem is implemented in the form of a file with the source program code and consists of the following points [23]:

1. Dimensions of the problem (number of phase coordinates, number of controls).
2. The initial values of the phase coordinates.
3. Imposed constraints on controls.
4. The right parts of the dynamic system with controls.
5. Analytical expressions for the derivatives of functions included in the problem statement. If they are absent, they are replaced by difference schemes.
6. Initial discretization of the problem, minimum and maximum sizes of the discretization grid.
7. The value of the required integration accuracy.
8. Maximum processor time for solving the problem.
9. Modes of logging, visualization, and fixation of results.

3 LOGICAL STRUCTURE OF THE SOFTWARE PACKAGE

The specialized software includes the following components:

1. A software framework that provides a uniform approach to the implementation of algorithms and includes basic functionality (organization of the process of numerical research, evaluation tools, random computing modules, etc.). The software package was developed using the C language and the GCC compiler; it functions under the operating systems of the Linux, Windows, and Mac OS families.
2. Algorithmic modules, including libraries of optimization algorithms. The implemented libraries of optimization algorithms include both multidimensional nonconvex optimization methods and algorithms for solving auxiliary problems of global univariate search (see, for example, [24–28]).
3. A collection of test problems with known solutions.
4. Tool and service modules.
5. Metacomponents.

The problem statement is formed in the c-file. After that, all the necessary components are compiled into a program designed to solve this particular mathematical formulation, at the same time, many parameters of the technological formulation can be changed online during the calculations. The result of the work of the obtained program will be a set of text files with data on the obtained solution, the course of the computational process and the processor time of the solution of the problem, statistics on multiple runs of the algorithms, and, if necessary, other data.

The functional subsystems of the developed software are described in more detail in the next section with an indication of their file compositions (in which files of the software package a particular module is implemented). The logical structure of the software package is shown in Figure 1.

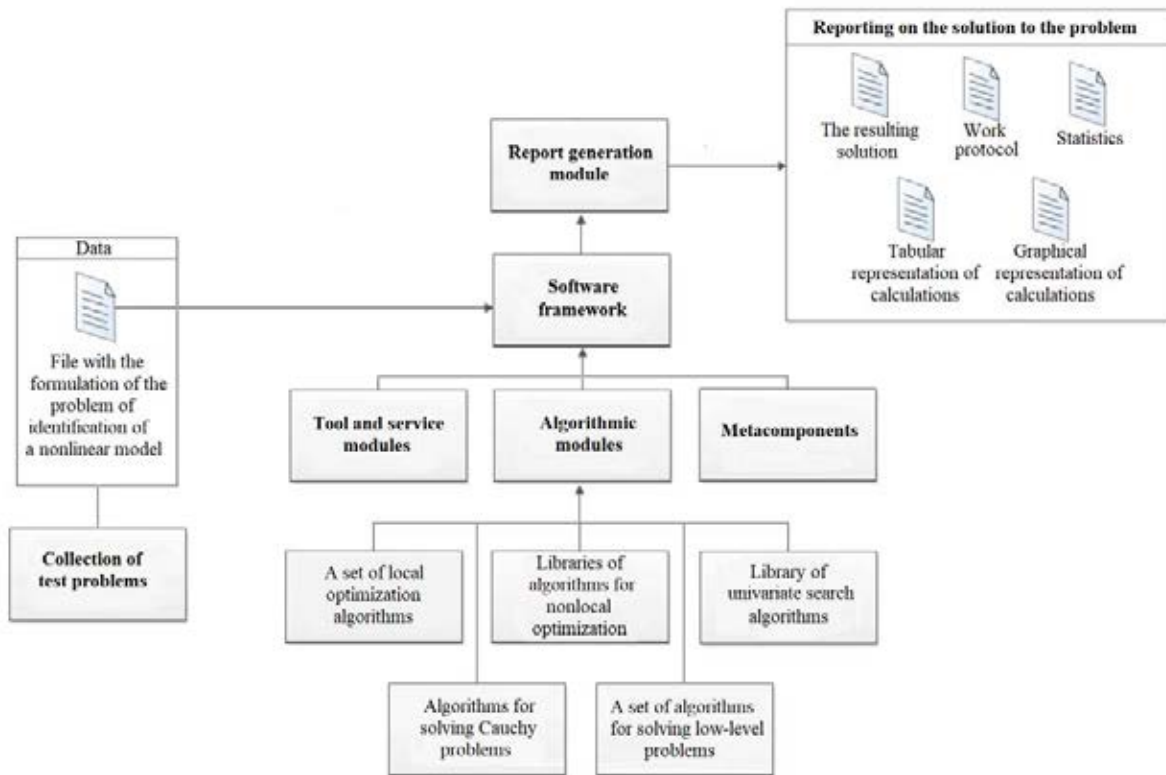


Figure 1: The logical structure of the developed software package

4 IMPLEMENTED FUNCTIONAL SUBSYSTEMS

Algorithmic modules of the software package include:

1. libraries of search algorithms for non-convex optimization [19]:
 - 1.1. library of algorithms based on one-dimensional global search methods: modifications of the “parabolas”, tunnel search, Powell, Rosenbrock, curvilinear search, partan, spherical search methods (parabolas.c, tunnel.c, powell.c, rosenbrock.c, curvilinear.c, partan.c, spheric.c);
 - 1.2. set of search algorithms that don’t use one-dimensional optimization methods: Luus–Jaakola, Rastrigin, tabu search, stochastic coverings, expert methods (luus.c, rastrigin.c, tabu.c, scover.c, expert.c);
 - 1.3. library of “bioinspired” algorithms: modifications of the genetic search, differential evolution, biogeography, particle swarm optimization, firefly, flower pollination, harmony search, teaching–learning-based optimization, cultural methods (ga.h, gapso.h, difevo.h, bbo.h, pso.h, firefly.h, flower.h, harmony.h, tlbo.h, cultural.h);
2. set of algorithms for solving auxiliary problems of one-dimensional optimization:
 - 2.1. library of univariate global search algorithms: modifications of the Evtushenko, Piyavsky, Strongin, “parabolas”, Brent, Lera–Sergeyev, “compressive search” methods, the combined algorithm based on “parabolas” and Strongin methods, Zhigljavsky–Zilinskas “probabilistic” P-algorithms (one_evtushenko.c,

one_piyavsky.c, one_strongin.c, one_parabol.c, one_brent.c, one_lera.c,
one_contraction.c, one_strongin_par.c, one_palgo.c);

2.2. implementation of the combined Brent method for one-dimensional local search (one_loc_brent.c);

3. set of multivariate local search algorithms: BFGS, L-BFGS, conjugate gradient methods, gradient decomposition method (bfgs.c, lbfgs.c, cgm.c, raider.c);

4. library of algorithms for solving Cauchy problems (cauchy.c);

5. collection of algorithms for solving auxiliary lower-level problems: generating pseudo-random numbers, calculating integrals, interpolating phase coordinates and table-valued controls, solving systems of linear equations, etc. (standart.c).

Service and tool modules of the software package consist of:

1. means of logging the computational process (main.c);

2. tools for creating control points of the computation process (main.c);

3. tools for calculating statistics on running algorithms (stats.h);

4. tabular and graphical means of visualization of calculation results (postproc.h);

5. means of accounting for available analytical formulas for gradients (grad.c);

6. means of verification of analytical formulas for derivatives (grad.c);

7. algorithms for differential estimation of derivatives (grad.c);

8. means for constructing a suboptimal sampling grid (service.c);

9. algorithms for estimation of errors of integration and sampling (service.c).

Metacomponents of the MEOPT software package are:

1. dialogue system that implements an interface for the user (main.c);

2. means of informing and consulting the user (inform.c);

3. means of automation and control of the problem setting (main.c);

4. algorithms for planning and automation of calculations (planner.c);

5. mechanisms for setting the parameters of algorithms (main.c).

Collection of test problems. To test the MEOPT software package, the authors of the project are developing a specialized collection of test problems characterized by different levels of complexity and simulating specific difficulties inherent in the studied classes of problems. The library of test problems includes both mathematical programming problems and problems of optimization of dynamical systems.

5 OPERATING MODES OF THE SOFTWARE PACKAGE

5.1 Available operating modes

The software package can function in three modes:

1. Interactive. In the interactive mode, the user is allowed to control the course of the computational process and to choose the algorithms for launching himself using the dialogue system.

2. Batch. In batch mode, the calculation process is carried out without the participation of the user, who generates in advance for the computer both a program statement of the problem and a set of instructions for execution.

3. Dynamic scheduler. In the dynamic scheduler mode, multi-method schemes are automatically generated based on the basic algorithms from the library. The developed version of a probabilistic single-level dynamic scheduler of computations is described further in the paragraph.

5.2. Probabilistic single-level dynamic computation scheduler for the global search

To effectively solve complex problems of global optimization, the use of one method in some cases is not enough. Any successful search strategy is based on a balance between global scanning of the feasible set and local refinement of the obtained approximations. The paper proposes a variant of a probabilistic dynamic scheduler of calculations for the global search problem, implemented based on the proposed optimization algorithms.

The algorithm of the “probabilistic” scheduler is as follows:

1. All algorithms are launched one by one from the same starting points for a certain time.
2. For each global search algorithm, an estimate is calculated: $\Phi^k = \Delta f_{\max}^k - \Delta f^k$, where $\Delta f^k = f_{\text{rec}}^k - f^*$, f_{rec}^k is the record value of the function found by the k -th algorithm, f^* is the global minimum value, $\Delta f_{\max}^k = \max_{1 \leq i \leq M} (\Delta f^i)$, M is the number of global search algorithms.
3. For each local search algorithm, an estimate ϕ^k is calculated using a formula similar to that specified in step 2.
4. Based on the estimates Φ^k obtained, one of the global search algorithms is selected and launched in a probabilistic way until the stop criterion is met.
5. Based on the obtained estimates ϕ^k , one of the local search algorithms is selected and launched in a probabilistic way until the stopping criterion is met to refine the solution found at step 4.
6. If the criterion for stopping the scheduler is met, it stops working; otherwise, it goes to step 4.

The planner training methodology is based on the sequential launch of all basic globalized algorithms from the same starting points, the calculation of efficiency estimates, defined as the difference between the record and the found algorithm value of the objective function and the probabilistic restart of the algorithms with the best estimates. In this case, for any of the basic algorithms, a nonzero restart probability remains, depending on the vector of its estimates summed up at iterations, in this case acting as a training model. Local search algorithms are used to periodically refine the record value of the objective function by a method, the choice of which also depends in a non-deterministic way on its success at previous iterations. The criterion for stopping the work of the scheduler is exceeding the specified time limit.

6 CONCLUSIONS

- The software package MEOPT for the numerical solution of multiextremal optimization problems was developed using the C language and the GCC compiler. It operates under Linux, Windows, and Mac OS operating systems.
- The libraries of optimization algorithms have been created, systematically divided into several subsets based on the genesis of the algorithm. All algorithms were implemented using uniform software standards and integrated into the MEOPT package.
- Three operating modes of the developed software package are available: interactive, batch, and dynamic scheduler modes.
- To date, prototypes of the main modules of the package have been completed.
- The technical testing of the first version of the software was carried out.

7 FUNDING

This research was funded by the Russian Foundation for Basic Research, project number 19-37-90065.

REFERENCES

- [1] V.S. Popov, R.P. Fedorenko, *The set of programs for the approximate solution of optimal control problems (application description)*. Keldysh Institute of Applied Mathematics, 1984.
- [2] Yu.G. Evtushenko, O.P. Burdakov, A.I. Golikov, V.G. Zhadan, M.A. Potapov, *DIOS dialogue complex. Section of nonlinear programming*. Computing Centre of the Academy of Sciences, 1982.
- [3] N.I. Grachev, A.N. Filkov, *Solution of optimal control problems in system DIOS*. Computing Centre of the Academy of Sciences, 1986.
- [4] A.I. Tyatyushkin, *Application software packages: experience of use*. Nauka, 1989.
- [5] R. Gabasov, A.I. Tyatyushkin, A.I. Zholudev, N.M. Erinchek, Application package “Mathematical programming of multidimensional problems”. *Algorithms and programs: Information Bulletin*, **2(71)**, 33-33, 1986.
- [6] A.Yu. Gornov, A.I. Zholudev, A.I. Tyatyushkin, N.M. Erinchek, *Application software packages. Development experience*. Nauka, 1983.
- [7] V.I. Gurman, V.A. Baturin, I.V. Rasina, *Approximate methods of optimal control*. Irkutsk University, 1983.
- [8] A.I. Zholudev, A.I. Tyatyushkin, N.M. Erinchek, Numerical optimization methods for controlled systems. *Izvestia of the Academy of Sciences of the USSR. Technical cybernetics*, **4**, 14-31, 1989.
- [9] A.Yu. Gornov, A.O. Divakov, *OPTCON software package for solving optimal control problems. User’s manual*. Irkutsk Computing Center, 1990.
- [10] A.S. Anikin, A.Yu. Gornov, A.N. Andrianov, Architecture and functional content of the OPTCON-M software package for optimization of the potentials of atomic-molecular clusters. *18th Baikal All-Russian Conference “Information and Mathematical Technologies in Science and Management”*. Part 3, Irkutsk, Russia, July 2-9, 2013.
- [11] J.T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, Philadelphia, 2010.
- [12] K.L. Teo, C. Goh, K. Wong, *A unified computational approach to optimal control problems*. John Wiley and Sons, New York, 1991.
- [13] O.V. Stryk, *User’s guide for DIRCOL: A direct collocation method for the numerical solution of optimal control problems, Vol. II*. Technical University of Munich, 1999.
- [14] C.S. Schweiger, C.A. Floudas, *MINOPT: A software package for mixed-integer nonlinear optimization*. Princeton University, 1996.
- [15] M. Gerdt, *User’s guide OC-ODE (version 1.4). Technical report*. University of Würzburg, 2009.

- [16] T. Hirmajer, E. Balsa-Canto, J.R. Banga, DOTcvpSB, a software toolbox for dynamic optimization in systems biology. *BMC Bioinformatics*, **10**, 199-213, 2009.
- [17] P. Kühn et al., *MUSCOD-II User's Manual*. University of Heidelberg, 2007.
- [18] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, **11**, 1-36, 2019.
- [19] P.E. Rutquist, M.M. Edvall, Propt-matlab optimal control software. *Tomlab Optimization Inc.*, **260**, 12-12, 2010.
- [20] P. Falugi, E. Kerrigan, E. Van Wyk, *Imperial college London optimal control software user guide (ICLOCS)*, Vol. 28. Department of Electrical and Electronic Engineering, Imperial College London, 2010.
- [21] V.M. Becerra, Solving complex optimal control problems at no cost with PSOPT. *2010 IEEE International Symposium on Computer-Aided Control System Design*, Yokohama, Japan, September 8-10, 2010.
- [22] A.V. Rao, Trajectory optimization: a survey. *Optimization and optimal control in automotive systems*, **1**, 3-21, 2014.
- [23] A.Yu. Gornov, *Computational technologies for solving optimal control problems*. Nauka, 2009.
- [24] P.S. Sorokovikov, Modifications of nonlocal one-dimensional search algorithms based on the Hölder condition. *Bulletin of the Buryat State University. Mathematics, computer science*, **4**, 40-56, 2019.
- [25] A.Yu. Gornov, P.S. Sorokovikov, T.S. Zarodnyuk, Computational technology for global search based on modified algorithm of the univariate nonlocal optimization. *Proceedings of the 6th International Workshop "Critical Infrastructures: Contingency Management, Intelligent, Agent-Based, Cloud Computing and Cyber Security" (IWCI 2019)*, **169**, 189-193, 2019.
- [26] A.Yu. Gornov, T.S. Zarodnyuk, A.S. Anikin, P.S. Sorokovikov, The stochastic coverings algorithm for solving applied optimal control problems. *18th International Conference on Mathematical Optimization Theory and Operations Research (MOTOR 2019)*, Ekaterinburg, Russia, July 8-12, 2019.
- [27] P.S. Sorokovikov, A.Yu. Gornov, Combined non-convex optimization algorithms based on differential evolution, harmony search, firefly, and L-BFGS methods. *IOP Conference Series: Materials Science and Engineering*, **1047**, 012077, 2021.
- [28] P.S. Sorokovikov, Library of non-convex optimization algorithms of various genesis. *63rd All-Russian Scientific Conference of MIPT. Applied Mathematics and Computer Science*, Moscow, Russia, November 23-29, 2020.