Choice of Spacecraft Control Contour Variant with Self-configuring Stochastic Algorithms of Multi-criteria Optimization

Maria Semenkina[1, 2], Shakhnaz Akhmedova[1], Christina Brester[1] and Eugene Semenkin[1]
[1]*Institute of Computer Sciences and Telecommunication, Siberian State Aerospace University, Krasnoyarskiy Rabochiy ave., 31, Krasnoyarsk, 660014, Russia*
[2]*Institute of Mathematics and Computer Science, Siberian Federal University, Svobodnyj ave. 79, Krasnoyarsk, 660041, Russia*

Keywords: Spacecraft Command-programming Control Contours Modelling, Markov Chains, Effective Variant Choice, Multi-objective Optimization, Evolutionary Algorithms, Self-configuration, Bio-inspired Intelligence.

Abstract: Command-programming control contours of spacecraft are modelled with Markov chains. These models are used for the preliminary design of spacecraft control system effective structure. Corresponding optimization multi-objective problems with algorithmically given functions of mixed variables are solved with a special stochastic algorithms called Self-configuring Non-dominated Sorting Genetic Algorithm II, Cooperative Multi-Objective Genetic Algorithm with Parallel Implementation and Co-Operation of Biology Related Algorithms for solving multi-objective integer optimization problems which require no settings determination and parameter tuning. The high performance of the suggested algorithms is proved by solving real problems of the control contours structure preliminary design.

1 INTRODUCTION

The synthesis of a spacecraft control systems is a complex and undeveloped problem. Usually this problem is solved with more empirical methods rather than formalized mathematical tools. Nevertheless, it is possible to model some subproblems mathematically and to obtain some qualitative results of computations and tendencies that could provide experts with interesting information.

We will model the functioning process of a spacecraft control subsystems with Markov chains. We explain all results with small models and then give illustration of large models that are closer to real system. The problem of choosing an effective variant for a spacecraft control system is formulated as a multi-objective discrete optimization problem with algorithmically given functions. In this paper, we use self-configuring evolutionary algorithms, Cooperative Multi-Objective Genetic Algorithm and Co-Operation of Biology Related Algorithms to solve the optimization problem.

The rest of the paper is organized in the following way. Section 2 briefly describes the modeled system. In Sections 3 we describe models for command-programming control contours. In Section 4 we describe optimization algorithms that have been used. In Section 5, the results of the algorithms performance evaluation on spacecraft control system optimization problems is given, and in the Conclusion section the article content is summarized and future research directions are discussed.

2 PROBLEM DESCRIPTION

If we simplify then we can describe the system for monitoring and control of an orbital group of telecommunication satellites as an automated, distributed, information-controlling system that includes on-board control complexes (BCC) of a spacecraft and the ground-based control complex (GCC) (Semenkin, 2012) in its composition. They interact through a distributed system of telemetry, command and ranging (TCR) stations and data telecommunication systems in each. BCC is the controlling subsystem of the satellite that ensures real time checking and controlling of on-board systems including pay-load equipment (PLE) as well as fulfilling program-temporal control. "Control contours" contain essentially different control tasks

Semenkina, M., Akhmedova, S., Brester, C. and Semenkin, E.
Choice of Spacecraft Control Contour Variant with Self-configuring Stochastic Algorithms of Multi-criteria Optimization.
DOI: 10.5220/0006009502810286
In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016) - Volume 1*, pages 281-286
ISBN: 978-989-758-198-4

from different subsystems of the automated control system. In this paper we will consider command-programming contours.

All contours are not function dependable and have many indexes that leads to many challenges during choosing an effective control system variant to ensure to all of the control contours. All this problems are multi-objective with criteria that cannot be given in the form of an analytical function of its variables but exist in an algorithmic form which requires a computation or simulation model to be run for criterion evaluation at any point.

In order to have the possibility of choosing an effective variant of such a control system, we have to model the work of all control contours and then combine the results in one optimization problem with many models, criteria, constraints and algorithmically given functions of mixed variables. We suggest using adaptive stochastic direct search algorithms (evolutionary and bio-inspired) for solving such optimization problems. To deal with many criteria and constraints successfully we just have to incorporate techniques, well known in the evolutionary computation community.

To support the choice of effective variants of spacecrafts' control systems, we have to develop the necessary models and resolve the problem of evolutionary algorithms (EA) and bio-inspired methods settings for multi-objective optimization.

3 COMMAND-PROGRAMMING CONTROL CONTOUR MODELLING

The main task of this contour is the maintenance of the tasks of creating of the command-programming information (CPI), transmitting it to BCC and executing it and control action as well as the realization of the temporal program (TP) mode of control (Semenkin, 2012).

Markov chains can be used for modelling this contour because of its internal features such as high reliability and work stability. That is why we are supposing that all stochastic flows in the system are Poisson. If we suppose that BCC can fail and GCC is absolutely reliable, then we can introduce the following notations: λ_1 is the intensity of BCC failures, μ_1 is the intensity of temporal program computation, μ_2 is the intensity CPI loading into BCC, μ_3 is the intensity of temporal program execution, μ_4 is the intensity of BCC being restored after its failure. The graph of the states for command-programming contour can be drawn as in Figure 1.

There are also five possible states for this contour (Semenkin, 2012):

1. BCC fulfills TP, GCC is free.
2. BCC is free, GCC computes TP.
3. BCC is free; GCC computes CPI and loads TP.
4. BCC is restored with GCC which is waiting for continuation of TP computation.
5. BCC is restored with GCC which is waiting for continuation of CPI computation.

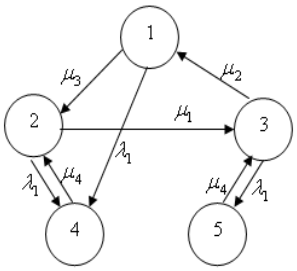

Figure 1: The states graph of the Markov chain for the simplified model of the command-programming control contour (Semenkin, 2012).

After solving the Kolmogorov's system:

$$P_1\cdot(\lambda_1+\mu_3) - \mu_2\cdot P_3 = 0,$$
$$P_2\cdot(\lambda_1+\mu_1) - \mu_3\cdot P_1 - \mu_4\cdot P_4 = 0,$$
$$P_3\cdot(\lambda_1+\mu_2) - \mu_1\cdot P_2 - \mu_4\cdot P_5 = 0,$$
$$P_4\cdot\mu_4 - \lambda_1\cdot P_1 - \lambda_1\cdot P_2 = 0,$$
$$P_5\cdot\mu_4 - \lambda_1\cdot P_3 = 0,$$
$$P_1 + P_2 + P_3 + P_4 + P_5 = 1.$$

we can calculate the necessary indexes of control quality for the command-programming contour:

1. $T = P_1/(\mu_2\cdot P_3)\rightarrow\max$ (the duration of the independent operating of the spacecraft for this contour);
2. $t_1 = (P_3+P_5)/(\mu_1\cdot P_2)\rightarrow\min$ (the duration of BCC and GCC interactions when loading TP for the next interval of independent operation of the spacecraft);
3. $t_2 = (P_2+P_3+P_4+P_5)/P_1\cdot(\lambda_1+\mu_3)\rightarrow\min$ (the average time from the start of TP computation till the start of TP fulfillment by BCC).

Optimization variables are stochastic flow intensities, i.e., the distribution of contour functions between BCC and GCC. If they are characteristics of existing variants of software-hardware equipment, we have the problem of effective variant choice, i.e., a discrete optimization problem.

Recall that obtained optimization problem has algorithmically given objective functions so before the function value calculation we must solve the system of equations.

But in real life GCC is not absolutely reliable, if we suppose the GCC can fail then we have to add the states when GCC fails while the system is in any state. We will not describe the meaning of all notion in details, recall that λ_i indicates the intensities of subsystems failures and μ_j indicates the intensities of subsystems being restoring by BCC (for PLE) or GCC (for all subsystems including itself) (Semenkin, 2012). Under the same conditions, the states graph for the command-programming contour consists of 96 states and more than 300 transitions and cannot be shown here.

So we need a reliable tool to solve so hard optimization problems, for example it can be some adaptive search algorithms.

4 OPTIMIZATION ALGORITHMS DESCRIPTION

There are many variants of evolutionary algorithms, which can be used for solving multi-objective optimization problems: Niched-Pareto Genetic Algorithm (NPGA) (Horn, 1994), Pareto Envelope-based Selection Algorithm (PESA) (Corne, 2000) and their modifications PESA-II (Corne, 2001), Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb, 2002), Strength Pareto Evolutionary Algorithm II (SPEA-II) (Zitzler, 2002), the Preference-Inspired Co-Evolutionary Algorithm with goal vectors (PICEA-g) (Wang, 2013).

In this paper we consider three algorithms: Self-configuring Non-dominated Sorting Genetic Algorithm II (SelfNSGA-II), Cooperative Multi-Objective Genetic Algorithm (CoMOGA) and Co-Operation of Biology Related Algorithms for solving multi-objective integer optimization problems (COBRA-mi).

4.1 Co-Operation of Biology Related Algorithms

Co-Operation of Biology Related Algorithms (COBRA) is a method for solving one-criterion unconstrained real-parameter optimization problems based on the cooperation of five nature-inspired algorithms (Akhmedova, 2013). This algorithm generates one population for each bio-inspired component algorithm, such as Particle Swarm Optimization (PSO) (Kennedy, 1995) for example. From some viewpoints these five algorithms have a similar behaviour and all of them are multi-agent algorithms for a stochastic direct search that makes almost impossible for end users choosing one of them for solving the problem in hand. COBRA is an algorithm with self-tuning of the population size that can increase or decrease depending on results of the work. Thus we have to choose only the maximum number of fitness function evaluations and initial number of individuals in population. After that population sizes can decrease for worst populations, increase for a winning algorithm and be constant when communicating with other algorithm.

COBRA performance was evaluated on the representative set of benchmark problems with 2, 3, 5, 10, and 30 variables (Akhmedova, 2013). Based on the test results we can say that COBRA is reliable on this benchmark and outperforms its component algorithms. We may conclude that COBRA can be used for solving our problem in hand.

COBRA has a multi-objective version (COBRA-m) (Akhmedova, 2015) with modified components. All these techniques were extended to produce a Pareto optimal front. So each component algorithm generates an archive of non-dominated solutions and a common external archive is created. The procedure of selecting the winning algorithm was changed by the modification of the fitness function when criteria are weighted by randomly generated coefficients (Akhmedova, 2015). The performance of COBRA-m was evaluated on the representative set of multi-objective problems and its usefulness and workability were established (Akhmedova, 2015).

Our problem in hand is an integer optimization problem. That is why real numbers are rounded to the nearest integers. Additionally, we make two modification of COBRA-m to tackle constraints: if individual falls beyond the set of possible values of the variables then

1. It is randomly regenerated to be inside limits (COBRA-mi), or alternatively
2. It is returned on the closest border (COBRA-mim).

Both variants are investigated in this study.

4.2 Self-configuring Evolutionary Algorithms

According to the research (Wang, 2013) NSGA-II is the most often used variant. NSGA-II is more effective than its predecessor (NSGA) in the sense of computational resources and the quality of the solutions (Abraham, 2005). Although its efficiency decreases with the growth of the criteria number we will implement it for solving our problem in hand because we have only three criteria.

Although NSGA-II is successful in solving many real world optimization problems (Abraham, 2005), its performance depends on the selection of its settings and tuning of its parameters. As well as for the genetic algorithm we need to choose variation operators (e.g. recombination and mutation) which are used to generate new solutions from the current population and some real-valued parameters of the chosen settings (the probability of recombination, the level of mutation, etc.). For reducing the role of this choice we have implemented a modification of NSGA-II transforming it into a self-configuring variant likely to SelfCGA from (Semenkin, 2012) which has demonstrated good performance for the integer optimization problem with one criteria and better reliability than the average reliability of the corresponding single best algorithm.

As it was done in (Semenkin, 2012) we use setting variants, namely types of crossover, population control and a level of mutation (medium, low, high) but we do not need to choose selection type because NSGA-II has the only one selection variant, namely the quick non-dominated sorting. Each of these has its own deployment probability distribution that is changed according to a special rule based on the operator productivity. In case of the genetic algorithm the ratio of the average offspring fitness obtained with this operator and the offspring population average fitness was used as the productivity of an operator. But in case of multi-objective optimization we have not just one function. That is the reason why we have implemented some modification in the operator of productivity evaluation such as the use of percent of non-dominated individuals which are generated by each operator type instead of average fitness value in SelfCGA. This introduced here algorithm we will refer as SelfNSGA-II.

4.3 Cooperative Multi-objective Genetic Algorithm

Another variant of self-adapting evolutionary algorithm for multi-objective problems can be described as a cooperation of several multi-objective genetic algorithm (MOGA). In our study an island model is applied to involve a few GAs which realize different concepts.

Generally speaking, an island model (Whitley et al., 1997) of a GA implies the parallel work of several algorithms. A parallel implementation of GAs has shown not just an ability to preserve genetic diversity, since each island can potentially follow a different search trajectory, but also could be applied to separable problems. The initial number of individuals M is spread across L subpopulations. At each T-th generation algorithms exchange the best solutions (*migration*). There are two parameters: *migration size*, the number of candidates for migration, and *migration interval*, the number of generations between migrations. Moreover, it is necessary to define the island model topology, in other words, the scheme of migration. We use the fully connected topology that means each algorithm shares its best solutions with all other algorithms included in the island model. The multi-agent model is expected to preserve a higher level of genetic diversity. The benefits of the particular algorithm could be advantageous in different stages of optimization.

In our implementation the NSGA-II, PICEA-g and SPEA2 are used to be involved as parallel working islands. This multi-agent heuristic procedure does not require additional experiments to expose the most appropriate algorithm for the problem considered. Its performance was thoroughly investigated on the set of test functions CEC2009 (Zhang., 2008). The results obtained demonstrated the high effectiveness (Brester, 2015) of the cooperative algorithm (CoMOGA) and, therefore, we also decided to apply it as an optimizer in the current problem.

4.4 Hybridisation with Pareto Local Search

The main idea of a Pareto local search is to find a local Pareto non-dominated point near of the start point. This algorithm changes the start point if only it has been dominated by some neighbour point. In this work we will use the steepest decent strategy and Hamming metrics for determination all point's neighbours in case of binary variables. Taking into account the properties of the problem in hand we can say that using the Pareto local search after global optimization techniques can give us a guarantee that point lays inside of Pareto set.

5 PROBLEM SOLVING RESULTS

First of all we evaluate performance of algorithms on the simplified models of command-programming control contours with 5 states. To choose an effective variant of the command-programming control contour we have to optimize the algorithmically given function with 5 discrete variables. The optimization space contains about $1.03 \cdot 10^6$ variants and can be enumerated with an exhaustive search within a reasonable time (60 minutes by Intel Core i5-4690K CPU 3.5 GHz). In such a situation, we know the real

Pareto front and can make an analysis. In all figures below the projection on the plane OTt_2 of the real Pareto front for problem with 5 states are presented by white points mean points. We can note that for this problem the criterion t_1 does not contradict two others and the Pareto front in other projections looks like horizontal (Ot_2t_1, OTt_1) or vertical (Ot_1t_2, Ot_1T) lines.

We use 600 for each algorithm. This means the algorithm examines 600 points of the optimization space, i.e. about 0.057% of it (less than 1 minute by the same CPU). We execute 20 runs of the algorithms and determine their usual (typical) behaviour (almost impossible to distinguish the results of two separate runs).

The usual behaviour of COBRA-mi is presented in Figure 2 (here and below black points mean the algorithm result). We can note that it gives points from the neighbourhood of Pareto front points which can be then reduced to Pareto front points with the help of the Pareto local search. The main problem for COBRA-mi became the necessity for some variables to catch exact border values. It is the main reason for introducing COBRA-mim.

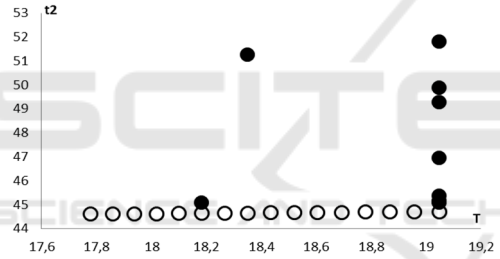

Figure 2: The projection of the Pareto front on the plane OTt_2 obtained with COBRA-mi.

The usual behaviour of COBRA-mim is depicted in Figure 3. We can note that it gives some points from the Pareto front. The main problem for COBRA-mim arises from criteria random weighting for resources redistributing. We prefer subpopulations with the best value of one criterion, i.e. different for each cycle. It means that we use additive convolution which cannot give all Pareto front points, but can catch the ends of the Pareto front.

The usual behaviour of SelfNSGA-II is presented in Figure 4. We can note that it gives about 60% points from the Pareto front that are uniformly distributed. However, SelfNSGA-II has some difficulties with catching ends of the Pareto front. It means that we can combine results of COBRA-mim and SelfNSGA-II and take almost all Pareto front points which can represent the whole front without essential loses, see the Figure 5. We can note that simple increasing of fitness function evaluations number do not lead to finding the whole Pareto front by single algorithm.

The usual behaviour of CoMOGA is presented in Figure 6. We can note that it gives points from the Pareto front which are well distributed. However, CoMOGA gave almost the same points as SelfNSGA-II, but not all of them, i.e. much fewer Pareto points.

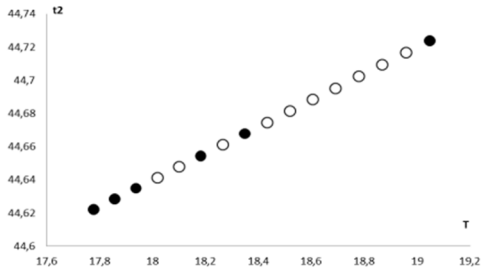

Figure 3: The projection of the Pareto front on the plane OTt_2 obtained with COBRA-mim.

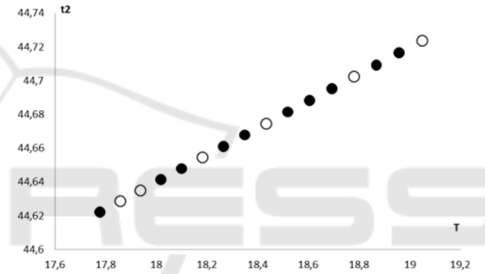

Figure 4: The projection on the plane OTt_2 of the Pareto front obtained by SelfNSGA-II.

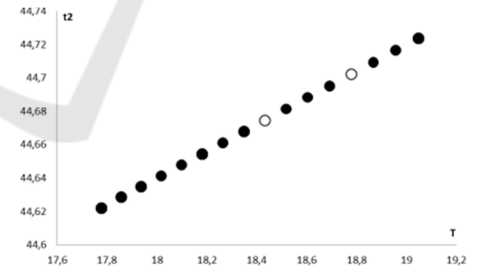

Figure 5: The projection of the Pareto front on the plane OTt_2 obtained with SelfNSGA-II and COBRA-mim.

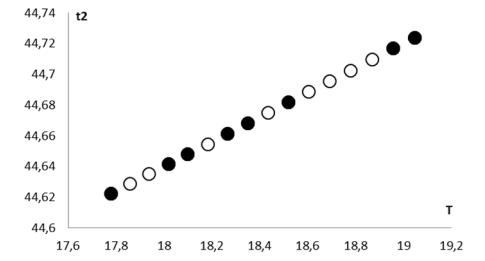

Figure 6: The projection of the Pareto front on the plane OTt_2 obtained by CoMOGA.

For the model of the command-programming control contour with 96 states and more than 300 transitions, we cannot give detailed information as we have done above. This problem has 13 variables and contains $4.5 \cdot 10^{15}$ points in the optimization space and an exhaustive search cannot be used for any reasonable time. The execution of our algorithms requires the examination of $1.76 \cdot 10^{-9}$% of the search space (80000 fitness function evaluations) and gives us as an answer only points from the Pareto front which have been verified with the Pareto local search. These points are uniformly distributed and look like the good representation of the Pareto front. However, certainly we cannot say at this stage of the research that all Pareto front points are determined.

6 CONCLUSIONS

In this paper, the mathematical models in the form of Markov chains have been implemented for choosing effective variants of spacecraft command-programming control contours. We focused on the multi-objective part of the problem and suggested using the Self-configuring Non-dominated Sorting Genetic Algorithm II, Cooperative Multi-Objective Genetic Algorithm and Co-Operation of Biology Related Algorithms for solving multi-objective integer optimization problems in such a situation because of their reliability and high potential to be problem adaptable. The high performance of the considered algorithms has previously been demonstrated through experiments with test problems and then in this paper it is validated by the solving hard optimization problems.

We suggested using three algorithms together as an ensemble for better representability of the Pareto front. These algorithms are suggested being used for choosing effective variants of spacecraft control systems as they are very reliable and require no expert knowledge in evolutionary or bio-inspired optimization from end users (aerospace engineers).

The future research includes the expansion into using the simulation models and constrained optimization problem statements.

ACKNOWLEDGEMENTS

This research is supported by the Ministry of Education and Science of Russian Federation within State Assignment № 2.1889.2014/K.

REFERENCES

Akhmedova, Sh., Semenkin, E., 2013. Co-Operation of Biology Related Algorithms. *In Proc. of Congress on Evolutionary Computation (CEC 2013),* pp. 2207–2214.

Akhmedova, Sh., Semenkin, E., 2015. Co-Operation of Biology-Related Algorithms for Multiobjective Constrained Optimization. *In ICSI-CCI 2015, Part I, LNCS 9140,* pp. 487–494.

Abraham, A., Jain, L., Goldberg, R., 2005. *Evolutionary multiobjective optimization: theoretical advances and applications*. New York: Springer Science, 302 p.

Brester, C., Semenkin, E., 2015. Cooperative Multi-Objective Genetic Algorithm with Parallel Implementation. *In Proc. of ICSI-2015,* pp. 471–478.

Corne, D., Knowles, J., Oates, M., 2000. The Pareto envelope-based selection algorithm for multiobjective optimization. *In PPSN VI, Parallel Problem Solving from Nature. Springer,* pp. 839-848.

Corne, D., Jerram, N., Knowles, J., Oates, M., 2001. PESA-II: Region-based selection in evolutionary multiobjective optimization. *In GECCO 2001, Proceedings of the Genetic and Evolutionary Computation Conference,* pp. 283-290.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *In IEEE Transactions on Evolutionary Computation 6 (2),* pp. 182-197.

Horn, J., Nafpliotis, N., Goldberg, D., 1994. A niched Pareto genetic algorithm for multiobjective optimization. *In CEC-1994,* pp. 82-87.

Kennedy, J., Eberhart, R., 1995. Particle Swarm Optimization. *In Proc. of IEEE International Conference on Neural networks, IV,* pp. 1942–1948.

Semenkin, E., Semenkina, M., 2012. Spacecrafts' Control Systems Effective Variants Choice with Self-Configuring Genetic Algorithm. *In Proc. of the 9th International Conference on Informatics in Control, Automation and Robotics, vol. 1*, pp. 84-93.

Wang, R., 2013. *Preference-Inspired Co-evolutionary Algorithms.* A thesis submitted in partial fulfillment for the degree of the Doctor of Philosophy, University of Sheffield, 231 p.

Yang, Ch., Tu, X., Chen, J., 2007. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. *In Proc. of the International Conference on Intelligent Pervasive Computing,* pp. 462–467.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., Tiwari, S., 2008. *Multi-objective optimization test instances for the CEC 2009 special session and competition.* University of Essex and Nanyang Technological University, Tech. Rep. CES-487.

Zitzler, E., Laumanns, M., Thiele, L., 2002. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. *In Evolutionary Methods for Design Optimization and Control with Application to Industrial Problems,* 3242 (103), pp. 95-100.