Parallel Version of Self-Configuring Genetic Algorithm Application in Spacecraft Control System Design

Maria Semenkina Siberian State Aerospace University Krasnoyarsky rabochy avenue, 31 660014, Krasnoyarsk, Russia +7-391-291-91-19 semenkina88@mail.ru

ABSTRACT

Technological and command-programming control contours of a spacecraft are modelled with Markov chains. These models are used for the preliminary design of spacecraft control system effective structure with the use of special DSS. Corresponding optimization problems with algorithmically given functions of mixed variables are solved with a special stochastic algorithm called self-configuring genetic algorithm that requires no settings determination and parameter tuning. The high performance of the suggested algorithm is proved by solving real problems of the control contours structure preliminary design.

Categories and Subject Descriptors

G.4 [**Mathematical software**]: – Algorithm design and analysis, efficiency

I.2.8 [Artificial intelligence]: Problem Solving, Control Methods, and Search – *Heuristic methods*

Keywords

Spacecraft Control Contours Modelling, Markov Chains, Effective Variant Choice, Complex Optimization Problems, Self-Configuring Genetic Algorithm, Island Model.

1. INTRODUCTION

One of the most difficult and underdeveloped problems is that of the synthesis of a spacecraft's control systems. This is currently solved with more empirical methods rather than formalized mathematical tools. Usually, the spacecraft control system design is a sophisticated process involving the cooperation of numerous experts and departments each having their own objectives and constraints. Nevertheless, it is possible to mathematically model some subproblems and to obtain some qualitative results of computations and tendencies that could provide interesting information for experts.

We suggest modelling the functioning process of spacecraft's control subsystems with Markov chains. The problem of choosing effective variant for a spacecraft's control system is formulated as a multi-scale optimization problem with algorithmically given functions. In this paper, we use sequential and parallel selfconfiguring genetic algorithm to solve the optimization problem.

*GECCO*¹³ *Companion*, July 6–10, 2013, Amsterdam, The Netherlands. ACM 978-1-4503-1964-5/13/07.

2. PROBLEM DESCRIPTION

The system for monitoring and control of an orbital group of telecommunication satellites is an automated, distributed, information-controlling system that includes in its composition onboard control complexes (BCC) of spacecrafts: telemetry, command and ranging (TSR) stations; data telecommunication subsystems; and a mission control center (MCC). The last three subsystems are united in the ground-based control complex (GCC). GCC interacts with BCC(s) through a distributed system of TCR stations and data telecommunication systems that includes communication nodes in each TCR, channels and MCC's associated communication equipment. BCC is the controlling subsystem of the spacecraft that ensures real time checking and controlling of on-board systems including pay-load equipment (PLE) as well as fulfilling programtemporal control. Additionally, BCC ensures the interactivity with ground-based tools of control. The control functions fulfilled by subsystems of the automated control system are considered to form subsets called "control contours" that contain essentially different control tasks. Usually, one can consider the technological control contour, command-programming contour, purpose contour, etc.

Each contour has its own indexes of control quality that cannot be expressed as a function of others. This results in many challenges when attempting to choose the effective control system variant to ensure high control quality with respect to all of the control contours. A multicriterial optimization problem statement is not the only problem. For most of the control contours, criterion cannot be given in the form of an analytical function of its variables but exists in an algorithmic form which requires a computation or simulation model to be run for the criterion evaluation at any point.

3. SELF-CONFIGURING GENETIC ALGORITHM APPLICATION IN SPACECRAFT CONTROL SYSTEM DESIGN

In order to have the possibility of choosing an effective variant of such a control system [1], 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 evolutionary algorithms (EAs) to solve such optimization problems as these algorithms are known as good optimizers having no difficulties with the described problem properties such as mixed variables and algorithmically given functions. The performance of EAs is essentially determined by their settings and parameters that requires time and computationally consuming efforts to find the most appropriate ones [2].

Copyright is held by the author/owner(s).

To support the choice of effective variants of spacecrafts' control systems, we have to develop necessary models and resolve the problem of EAs settings. The main idea of the approach used in this paper relies to automated selecting and using existing algorithmic components. That is why our algorithms might be called as self-configuring [3] ones. The self-configuring genetic algorithms (SelfCGA) uses binary solution representation, five selection operator types (fitness proportional, rank-based, and tournament-based with three tournament sizes), six crossover operator types (one-point, two-point, as well as equiprobable, fitness proportional, rank-based, and tournament-based uniform crossovers [3]) and three mutation levels (medium, low and high) during one iteration. Each of these has its own probability to be used. During the initialization all probabilities are equal and they are changed according to a special rule through the algorithm's execution.

The execution time of genetic algorithms (GA) application in the real world problem solving is their problem too, as in the case of a complex problem GA requires a greater number of individuals and generations to obtain good results. At the same time, in solving complex practical problems the calculation of a single objective function value can take a lot of time.

In our case it is necessary to solve equations system with many tens of equations. The main solution of this problem is the GA parallelization that can be done in two ways: simple distribution of individuals' fitness evaluation among many computing cores of usual multicore computer and the application of island-based GA framework [4].

The first approach admits the SelfCGA speed up proportionally to the number of cores without advantages in performance (solution quality and reliability) and serves mainly as a reference base for the second approach. The second approach is the 8 cores based island SelfCGA with clique connection graph between core populations which allow migration of the 5% of the best individuals after every 10% of the whole amount of generations number. Every population on the computing core evolves independently between migration time points. It is expected that this approaches will give not only algorithm speed up but also a positive effect on the performance.

The performance of a conventional "single" GA has been compared with the proposed SelfCGA on the 14 usual test problems for GA [5]. Saying "single" algorithm, we mean the group of algorithms with the same crossover operator but with all variants of other settings. After 1000 runs and statistical processing of results (see Table 1), the following observations were found in terms of algorithm reliability.

1 adie 1. Algorithm reliadilities averaged over 14 test broblem	Table	1. Algorithm	reliabilities	averaged or	ver 14 te	st problems
---	-------	--------------	---------------	-------------	-----------	-------------

Algorithm	Average	Worse	Best
"Single" best GAs	0,888	0,834	0,932
SelfCGA	0,928		
"Single" best GAs + parallel	0,907	0,861	0,949
SelfCGA+ parallel	0,963		

Analyzing results, related to sequential SelfCGA, we can see that SelfCGA demonstrates better reliability than the average reliability of the corresponding single best algorithm but worse than the best one. Additionally, we can observe that parallel SelfCGA demonstrates higher reliability than sequential one in all cases. It means that we may use the SelfCGA in real world problems solving.

First of all we evaluate its performance on the simplified models of technological and command-programming control contours with 5 states. Then we have to evaluate the performance of the suggested algorithm on generalized models which have much higher dimensions. The optimization model for the technological control contour has 11 discrete variables. The last problem has the model of the command-programming control contour with 96 states and more than 300 transitions. Results have shown that the SelfCGA outperforms all alternative algorithms for all problem statements and it's parallel variant outperforms sequential one. For example, reliabilities for technological control contour model with 40 states for the best alternative algorithm, sequential SelfCGA and parallel SelfCGA are equal 0.93, 0.96, 0.99, respectively.

We have no place to go into the details with estimations of parallel algorithms speed up. Nevertheless, we can remark that simple parallelization of SelfCGA on the 8 core computing system accelerates the execution in almost 8 times. It can be explained if we realise that any fitness evaluation is the time consuming process comparing with algorithm operations. We have also to remark that island model SelfCGA additionally accelerates execution because of the synergetic effect of the cooperation of independently evolving populations. This effect gives also positive impact on the algorithm performance.

4. CONCLUSION

The parallel self-configuring genetic algorithm has been suggested to be used for choosing effective variants of spacecraft control systems as it is very reliable and requires no expert knowledge in evolutionary optimization from end users (aerospace engineers).

5. REFERENCES

- [1] Semenkin, E., Semenkina, M. 2012. Spacecrafts' Control Systems Effective Variants Choice with Self-Configuring Genetic Algorithm. In: Ferrier, J.-L., Bernard, A., Gusikhin, O. and Madani, K. (eds), Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics, volume 1, 84-93.
- [2] Eiben, A.E., Hinterding, R., and Michalewicz, Z. 1999. *Parameter control in evolutionary algorithms*. In: IEEE Transactions on evolutionary computation, 3(2). IEEE Press Piscataway, NJ, USA, 124-141.
- [3] Semenkin, E., Semenkina, M. 2012. Self-Configuring Genetic Algorithm with Modified Uniform Crossover Operator. In: Tan, Y., Shi, Y., and Ji, Z. (Eds.): Advances in Swarm Intelligence, ICSI 2012, Part I, LNCS 7331. Springer, Heidelberg, 414–421.
- [4] Lardeux, F., Goëffon, A. 2010. A Dynamic Island-Based Genetic Algorithms Framework. In: Simulated Evolution and Learning, Lecture Notes in Computer Science 6457. Springer, Heidelberg, 156-165.
- [5] Finck, S., Hansen, N., Ros, R., and Auger, A. 2009. *Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions*. Technical Report 2009/20, Research Center PPE.