STATE RESEARCH CENTER OF THE RUSSIAN FEDERATION
CONCERN **CSRI ELEKTROPRIBOR**, JSC

CO-SPONSORED BY:

RUSSIAN FOUNDATION FOR BASIC RESEARCH

INTERNATIONAL PUBLIC ASSOCIATION - ACADEMY OF NAVIGATION AND MOTION CONTROL (ANMC)

NATIONAL RESEARCH UNIVERSITY ITMO, RUSSIA

AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS (AIAA)

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS - AEROSPACE AND ELECTRONIC SYSTEMS SOCIETY (IEEE - AESS)

INSTITUT FRANÇAIS DE NAVIGATION (IFN)

DEUTSCHE GESELLSCHAFT FUER ORTUNG UND NAVIGATION (DGON)
GERMAN INSTITUTE OF NAVIGATION

JOURNAL GYROSCOPY AND NAVIGATION

27-29 MAY 2019

26th SAINT PETERSBURG INTERNATIONAL CONFERENCE ON INTEGRATED NAVIGATION SYSTEMS

STATE RESEARCH CENTER OF THE RUSSIAN FEDERATION
CONCERN **CSRI ELEKTROPRIBOR**, JSC

STATE RESEARCH CENTER OF THE RUSSIAN FEDERATION
CONCERN CSRI ELEKTROPRIBOR, JSC

26th SAINT PETERSBURG INTERNATIONAL CONFERENCE ON INTEGRATED NAVIGATION SYSTEMS

PROCEEDINGS

27 – 29 May 2019

Saint Petersburg, Russia

CO-SPONSORED BY:

- **RUSSIAN FOUNDATION FOR BASIC RESEARCH**
- **INTERNATIONAL PUBLIC ASSOCIATION – ACADEMY OF NAVIGATION AND MOTION CONTROL (ANMC)**
- **NATIONAL RESEARCH UNIVERSITY ITMO, RUSSIA**
- **AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS (AIAA)**
- **INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS – AEROSPACE AND ELECTRONIC SYSTEMS SOCIETY (IEEE AESS)**
- **INSTITUT FRANÇAIS DE NAVIGATION (IFN)**
- **DEUTSCHE GESELLSCHAFT FUER ORTUNG UND NAVIGATION (DGON)
GERMAN INSTITUTE OF NAVIGATION**
- **JOURNAL GYROSCOPY AND NAVIGATION**

In the present publication the plenary and poster papers of the 26th Saint Petersburg International Conference on Integrated Navigation Systems (27 – 29 May 2019) are presented.

*The poster papers are marked with *.*

Editor-in-Chief

Academician of RAS
Vladimir G. Peshekhonov

Dear authors,

please, look through your paper. If there are any misprints, mistakes etc., please notify the editorial office at: **editor@eprib.ru** before **10.06.19 inclusive**.

Otherwise the paper will be accepted by default and loaded into the citation database as it is presented in the Proceedings.

The Use of a Genetic Algorithm for Parameter Adjustment of the Multi-Hypothesis Aircraft Tracking Algorithm*

Dmitrii Bedin
Dynamical Systems department
Krasovskii Institute of Mathematics and Mechanics,
Ural Branch, Russian Academy of Sciences
Ekaterinburg, Russia
bedin@imm.uran.ru

Alexey Ivanov
Dynamical Systems department
Krasovskii Institute of Mathematics and Mechanics,
Ural Branch, Russian Academy of Sciences
Ekaterinburg, Russia
iagsoft@imm.uran.ru

***Abstract*—In previous studies, we developed a multi-hypothesis algorithm for the aircraft tracking, which shows good performance in the processing of real and model data. The algorithm operation depends on a large number of parameters. It is quite difficult to detect the influence of some parameters on the results. In this situation, it is justified to use a genetic algorithm to adjust the parameters.**

***Index Terms*—Genetic algorithms, Radar tracking**

I. Introduction

Previously, the authors developed the multi-hypothesis algorithm [1]–[3] for the aircraft tracking. The algorithm was tested on simulated and real data and showed good performance. However, the algorithm operation depends on a large number of parameters, and the impact of their changes on the results of computation is very confusing. Initially, the values of most parameters were chosen based on some empirical considerations. To improve the performance of the algorithm, it is necessary to make the parameter setting, which should be carried out automatically, with testing the algorithm on simulated data (machine learning).

Parameter setting methods that use partial derivative values cannot be applied to all parameters of a multi-hypothetical algorithm. This is because, for some parameters, the calculation of derivatives is very laborious both in terms of writing the required code and in terms of computational resources. But the other parameters are integer, hence, the derivative with respect to them can exist only in some special generalized sense.

It is reasonable to use a genetic algorithm for parameter adjustment. Relatively easy for programming, the genetic algorithm requires a lot of computing resources. But this requirement is not critical due to the computer cluster available at the Institute of Mathematics and Mechanics UB RAS [4].

II. Multi-hypothesis aircraft tracking algorithm

The basis of the multi-hypothetical algorithm is the construction and maintenance of a bundle of trajectories, each of them is a variant of the aircraft motion combined with the available measurements, observation history, and restrictions on the dynamics [1].

Upon receipt of the next measurement, the algorithm produces the development of the bundle (continuation, branching, *etc.*) in accordance with the coordinates and the instant of measurement. Then the construction of a position estimation of the aircraft is performed at the measurement time. Next, the bundle is pruned and the algorithm waits for the next measurement.

The algorithm proved to be quite good, especially in the cases of the presence of outliers. Outliers are rare measurements with a high level of error [2], [3].

The disadvantage of the algorithm is a large number (several tens) of tuning parameters whose effect on the algorithm operation is not always obvious. For example, when calculating the main criterion for the correspondence of the trajectory to the available measurements, we use some specific distance function between the measurement and the point on the trajectory. This function has two parameters: “dead zone value” and “far zone value”. The value of the main correspondence criterion is used, in particular, when the bundle is pruned. But it is very difficult to consider analytically the impact of the values of the mentioned parameters on the algorithm operation even at the one-step stage of the algorithm.

III. Genetic algorithm

This algorithm is based on the simulation of the natural evolution process [5] with inclusion of elements of some direct search method.

The program of the genetic algorithm is written in MATLAB and the computing part (the multi-hypothetical algorithm for the aircraft tracking) is written in Free Pascal.

A. Data structures

The structures are designed as classes in MATLAB.

Class “Parameter” (“gene”). The gene describes one of the parameters of the computing program of the multi-hypothetical algorithm for the aircraft tracking.

This work was supported by the Russian Foundation for Basic Research under project no. 18-01-00410.

Class “Individual” is designed to store the genotype (array of elements of the class “Parameter”) of one creature of a population. In addition to the genotype, the class “Individual” contains: properties related to the quality of the creature, the flag “immortality”, the maximum age index of the creature, the current age of the creature, and the parameters describing the creature parents.

Class “Population” stores an array of creatures of a population and certain parameters related to the overall quality of the population.

B. Algorithm initialization

The algorithm is initialized starting from one creature. After the formation of one creature, a population is formed on the basis of mutation: a certain number of creatures are formed for which the values of the parameters differ from those of the first one.

C. Calculation of new creatures

The new creatures are calculated by the multi-hypothesis aircraft tracking program [1]–[3].

We set an ideal model trajectory that represents some true aircraft motion and contains sections of various types of movement characteristic of civil aircraft. On the basis of this trajectory, a package of measurement tracks with a different implementation of random errors is constructed.

Each creature has a different genotype, *i.e.*, from the point of view of the tracking program, a set of parameters. Accordingly, estimations of the current aircraft location (which the program will form when using the parameters) will be different.

After the calculation of the package of measurement tracks, the program writes the result file (the reconstructed trajectories). Based on the file, the MATLAB script generates multiple quality ratings (criteria) of a specific creature. Among these criteria are the maximum deviation of the restored evaluation from the true position for all measurements of all trajectories and the average over all the trajectories of the standard deviation of the estimated trajectory from the true one.

The fitness function h is determined on the basis of the criteria values as the weighted average of the individual criteria. The sense of the fitness function is that it is a generalized deviation of the restored trajectory from the true one.

For each of the criteria, the least value is the best. The dimension of all criteria is meters.

In addition, on the basis of the criteria, creatures are assigned the sign of immortality: if the population minimum of some criterion is implemented for a certain creature, then this creature is marked with the “immortality” flag.

Note that the number of immortal creatures can be less than the criteria number (one creature implements the minimum of several criteria) as well as more than the criteria number (several creatures have a criterion equal to the minimum for the population).

The maximum assigned age for each creature is calculated based on the basic criterion h as follows

$$H_{\min} = \min_A h, \quad H_{\max} = \max_A h,$$

$$T_L(h) = 1 + M_A \frac{H_{\max} - h + 1}{H_{\max} - H_{\min}}.$$

Here, A is the set of all creatures in the population, the constant M_A is the normal maximum age, which is a parameter of the algorithm.

D. Selection

The selection destroys “old” creatures (with the age greater than T_L), except those with the “immortality” flag. If, after the destruction of “old” creatures, the number of creatures exceeds the designated maximum number, then the “elder” creatures for which the difference between the current age and the maximum designated age is small are under destruction.

E. Crossover

Any two creatures aged older than 2 can be subjected to the crossover. Pairs of creatures are formed with the help of a random number generator. A pair of creatures forms a new creature so that each gene of the new creature is inherited from either the first or the second parent; the order of inheritance is chosen by a random number generator. If the resulting descendant genotype is a clone of one of the parents, then the creature based on such a genotype is not formed.

F. “Directed reproduction”

This type of inheritance introduces the element of direct search into the genetic algorithm. An attempt is made to generate guidedly a descendant genotype with the best (compared to the parents) value of the fitness function h. A pair of creatures with different values of h is used.

The idea of the algorithm is illustrated in Fig. 1. Here, A is the genotype of the creature with a larger value of the function h and B is the genotype of the individual with a smaller (better) value of the function h. If we assume the continuous dependence of the function h on the genotype, then dependencies are very likely to be close either to the solid curve (the minimum between A and B) or to the dotted curve (there is the minimum on the other side from B opposite to A) in the left figure.

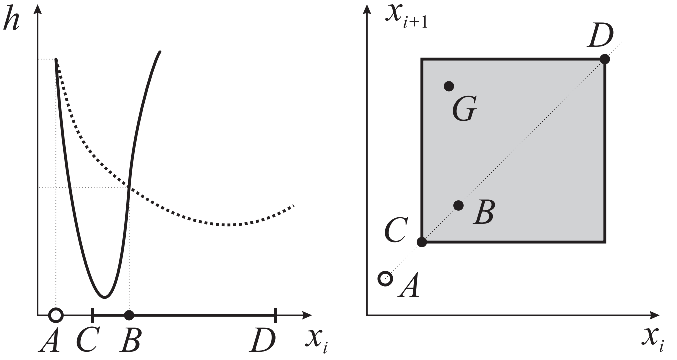

Fig. 1. Schematic illustration of the idea of the method “Directed reproduction”: on the left, the projection is on the plane $x_i \times h$; on the right, the projection is on the plane $x_i \times x_{i+1}$.

The point C is chosen in the middle of the segment AB. The point $D = 3B - 2A$ is the continuation of the vector AB

in the direction towards "the other side of B". The genotype of the descendant G is randomly selected on the parallelotope CD.

Such inheritance works well in the case when the genotype of the majority of creatures in the population is far enough from the local minimum.

G. Mutations

Immortal creatures mutate on each generation of the population, while other creatures mutate with a low probability.

If the maximum assigned age for a creature is less than the current age, then the creature undergoes the normal mutation: the value of each gene is randomly changed by an amount not greater than the current step of variation of this gene.

If the creature age exceeds the maximum assigned age (this is possible for immortal creatures), then the creature is subjected to either micromutations or macromutations.

A micromutation is the mutation when the value of each gene is changed by a random value of two orders of magnitude smaller than the current variation step of the gene.

In the case of macromutation, the magnitude of only one randomly selected gene is changed and the value of this gene is changed by a random value of two orders of magnitude greater than the current step of the gene variation. In some cases of the macromutation, the mutated gene is selected randomly over the whole region of possible values.

This behavior is aimed at reviving the genetic diversity in the event of a possible "stagnation" of the population.

IV. Calculation results

Several computational experiments were made to find the best parameters of the multi-hypothesis algorithm. We give a description of one of them.

As an ideal model trajectory, we take the trajectory with a duration of 1100 s that consists of sections of steady motion and transition sections, for which the maximum estimate errors are defined in the normative materials [6]. The package of model trajectories consisted of 16 trajectories with standard deviations of 70 m, among them 10 trajectories contained no outliers and 6 trajectories had outliers (on the average, every twentieth measurement had a noise with the enhancement factor of 5).

The variant with 19 variable parameters (19 genes) was calculated.

The calculations were performed using the "Uran" supercomputer of IMM UB RAS [4], where up to 40 computing cores were simultaneously used; the duration of the calculations was a little more than two days. During this time, 419 generations have changed in the genetic algorithm.

Fig. 2 shows the development of the fitness function h from generation to generation. Each creature of the population is represented by some marker (the shape and color of the marker depends on the method of formation of the creature). The figure displays a region with small values of h.

Fig. 3 illustrates the evolution of the population using the example of the `w0_penalty_cir` gene. The initial value of the gene was 0.01; for the first few tens of generations, most creatures had values of this gene in the range $(0, 0.04)$. In the section between the generations 100 and 150, the gene values were more or less evenly distributed over the segment of possible values. After the generation 150, most of the creatures began to have values of the gene in the range $(0.32, 0.36)$. In the last generation, 0.3498 gives the best value of the fitness function h.

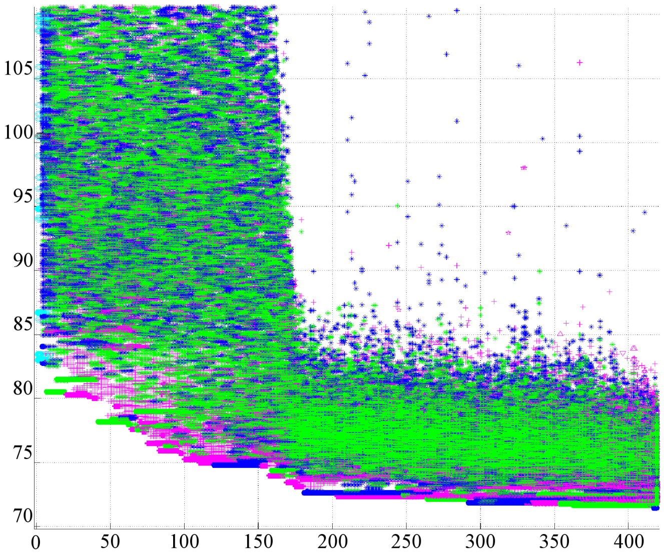

Fig. 2. Graph of changes of the fitness function h in the population depending on the generation. The vertical axis is the value of h, m. The horizontal axis is the generation number.

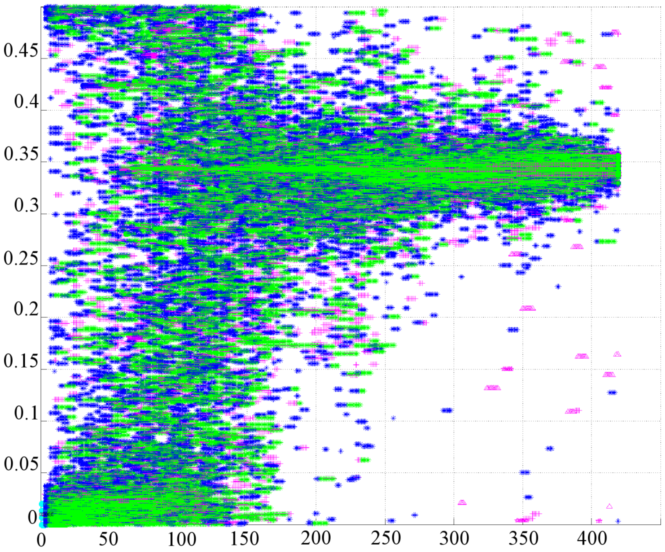

Fig. 3. Changing in the distribution of the `w0_penalty_cir` gene values in the population depending on the generation

Fig. 4 shows how the multi-hypothesis algorithm improves its work when using a parameter vector optimized by the genetic algorithm. We used a package of trajectories obtained on the basis of the same ideal model trajectory, but consisting of 100 trajectories with a different implementation of random errors, among them 10 trajectories having outliers. The graph of the standard deviation of the reconstructed points from the true motion along the transverse channel is given. The wide solid line shows the results for the optimized vector of parameters. Results for the original (non-optimized) parameter vector are shown by the dotted line. The thin solid line shows the deviation graph for the measurement tracks.

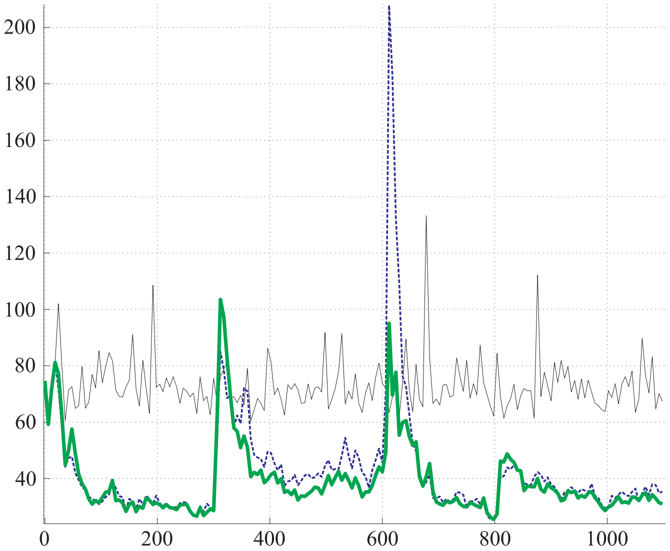

Fig. 4. Comparison of processing the model trajectories for the original and improved parameters. A package of 100 trajectories was used, which does not coincide in noise with the main computational package that the genetic algorithm worked on.

Acknowledgment

The authors thank NITA (New Information Technologies in Aviation, LLC) for the formulation of the problem and the discussion of the obtained results.

Our study was performed using the “Uran” supercomputer of the Krasovskii Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences.

References

[1] A. A. Fedotov and A. G. Ivanov, “Multi-hypothesis tracking algorithm for aircraft trajectory,” in Systems Analysis: Modeling and Control, Abstracts of the International Conference in Memory of Academician Arkady Kryazhimskiy, Ekaterinburg, Russia, IMM UB RAS, 2016, pp. 44–46.

[2] D. A. Bedin, A. A. Fedotov, and A. G. Ivanov, “Recovering an aircraft trajectory by using the detection of the motion type,” in 25th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2018, St. Petersburg, CSRI Elektropribor, 2018, pp. 127–130.

[3] A. G. Ivanov, D. A. Bedin, A. A. Fedotov, and S. A. Ganebnyi, “Fixed Types of Motion in Aircraft Trajectory Recovering,” IFAC-PapersOnLine, 2018, vol. 51, iss. 32, pp. 889–894.

[4] Multiple-Access Computing Center of IMM UB RAS “Supercomputer Center IMM UB RAS” (in Russian). [Online]. Available: http://parallel.uran.ru/node/419

[5] M. Mitchell, An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1996.

[6] SUR.ET1.ST01.1000-STD-01-01: EuroControl standart document for radar surveillance in en-route airspace and major terminal areas. Edition 1.0, 1997. [Online]. Available: https://www.eurocontrol.int/publications/eurocontrol-standard-radar-surveillance-en-route-airspace-and-major-terminal-areas