MULTI OBJECTIVE OPTIMIZATION ALGORITHMS AND SAFETY NETS IN ATC SYSTEM

Liudas Mašnauskas	habil. dr. prof. Jonas Stankūnas
Vilnius Gediminas Technical University	Vilnius Gediminas Technical University
Saulėtekio al. 11, LT–10223 Vilnius, Lithuania	Saulėtekio al. 11, LT–10223 Vilnius, Lithuania
masnauskas.l@ans.lt	jonas.stankunas@vgtu.lt

Abstract. False safety nets alerts are highly undesirable in ATC environment. Safety nets performance and parameters tuning forms multi objective optimization problem. Multi objective optimization algorithms are used for engineering purposes. Possibility of using multi objective optimization algorithms for improving safety nets alarms performance and minimization of false alarm rate is discussed.

Keywords. optimization, safety nets, genetic algorithm.

1 Introduction

European Union moves towards standardization of ground based Safety Nets (STCA, MSAW, APM and APW). All European air navigation service providers (ANSP) are expected to implement a system which at least conforms to the minimum requirements laid out in the Eurocontrol specification [6]. The main contributors to Safety Nets warning calculation are:

-Primary and secondary radar data,
-Aircraft performance data,
-Flight plan data,
-System variable parameters,
-Internal calculation algorithms.

Safety nets system shall predict aircraft position up to 320 seconds in advance. The prediction is based on data sources mentioned above and system variable parameters. System variable parameters are tuned by system administrator who has great experience in air traffic control. The objective of this work is to minimize number of false warnings and maximize number of true warnings. So we can state that there are two objectives -minimize negative false warnings and minimize negative true warnings. The optimization parameters are system variable parameters. However these parameters can be changed in some limits. So optimization problem has constraints. We can see that problem has all needed attributes for multi-objective optimization.

1.1 Safety nets in Air Traffic Control system

Eurocontrol [1] has produced recommendations for the reference Safety nets systems. The recommended STCA algorithm has three main steps:

-Coarse filter;
-Fine filter;
-Alert confirmation.

The whole algorithm is presented in picture below:

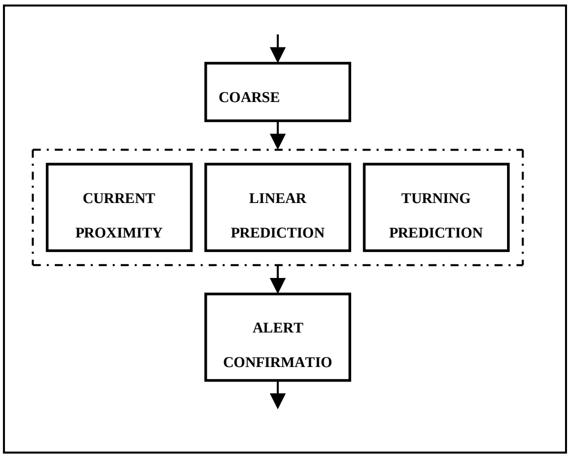

Figure 1: Recommended Eurocontrol algorithm

The coarse filter the pairs of system tracks which could be in potential conflict in the nearest future. The calculations are made in horizontal and vertical planes. The pairs which do not produce conflict are simply disregarded. In this case system resources are saved. The coarse filter output is the system track pairs which could be in conflict.

Linear prediction filter searchers the coarse filter output for the system track pairs which could be in conflict for the defined time period in advance. The calculations of three dimensional vector are made using linear prediction. However there are situations when linear filter is not effective. The case could be when aircraft are moving towards each other very slowly. In this case the current proximity filter is used. This filter is constantly comparing horizontal and vertical distances with with certain system parameters. When aircraft makes maneuver the linear prediction filter is not effective too. System uses turn prediction filter. Prediction filter notices that aircraft is in maneuver and predicts its position using aircraft performance parameters.

The last step is alert confirmation. Its purpose is:

- Decide if system tracks are in real conflict and produce warning immediately;
- Postpone the warning if radar data is not accurate enough;
- Postpone the warning if the conflict was produced by short-term situation;
- Decide if the warning has to be produced or wait for the one more cycle expecting that situation is going to change.

It needs to be said that Safety nets system performance depends on airspace configuration, system parameters tuning, traffic complexity and intensity. The traffic intensity is always growing. It means that more systems track pairs shall be analyzed and the rate of false warnings could increase to unacceptable levels. The example of false warnings is presented in table below:

Table No.1

False warnings

TYPE	DATE	FDR_1	FDR_1_COORDINATES	FDR_2	FDR_2_COORDINATES	DISTANCE
START STCA	Sat Aug 11 08:38:21 2012	BTI34H	5429N02507E	LYLAD	5427N02518E	20.6598 km
START STCA	Sat Aug 11 08:54:19 2012	WZZ739	5429N02510E	LYLAD	5435N02520E	21.0856 km
START STCA	Sat Aug 11 09:04:18 2012	SDM6754	5357N02251E	Track with SSR code 4323	0	0 km
START STCA	Sat Aug 11 09:12:22 2012	BTI34G	5439N02518E	LYLAD	5438N02520E	4.06744 km
START STCA	Sat Aug 11 09:28:51 2012	LYLAD	5435N02521E	AUA833V	5427N02508E	27.5729 km

The period was taken for two hours. All these warnings are false. The first column represents the type of warning. The second column shows the time of the warning. The FDR1 and FDR2 indicate two aircraft pair. Coordinates enables to calculate the distance in horizontal plane.

1.2 Problem definition

The safety nets optimization is understandable as maximization of true alerts and minimization of false alerts. However we cannot improve one without adversely affecting another as the same system parameters are used for calculations. For example if we prolong the prediction time we can got more false alerts. If we shorten prediction time we could miss the true alerts. We could construct the abstract curve which represents our task:

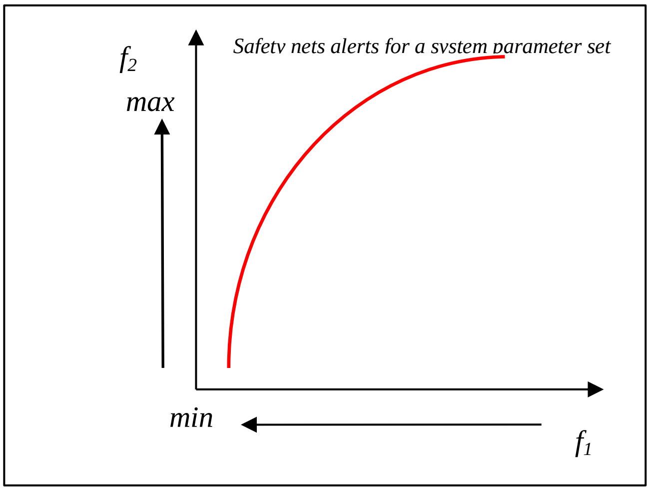

Figure 2: Optimization task

So we can say that we have multi objective optimization problem. Usually multi-objective optimization gives more than one solution. In this case we need some compromise. Optimization can be defined as a search in a defined set for the element with which criterion value should be minimized (in other cases maximized). Very often the analyzed set is Euclidian set. So we could speak about the search of a vector (parameter) for which the criterion value is minimum. Parameter vector $X \in Rn$ components x_i, $i = 1, \ldots, n$, are called problem variables and function defining criteria *f(x)* is called objective function [6]. But first of all we need to clarify which criteria shall be optimized, which parameters could be changed and in what limits. In other words we need to define the vector of variables *X*, objective function *f(X)* and allowed area *A*. After application of optimization method the optimal parameter vector X_0 is got. In common case of multi-objective optimization problem which has *m* objective functions:

$$f_1(x)\ f_2(x), \ldots, \ f_m(x) \qquad (1)$$

Is formulated like this: find such parameter vector:

$$X = (x_1, x_2, \ldots, x_d) \qquad (2)$$

which minimizes (maximizes) criterion vector:

$$F(X) = (f_1(x)\ f_2(x), \ldots, \ f_m(x)) \qquad (3)$$

Parameter vector $x = (x_1, x_2, \ldots, x_n)$ is chosen from a particular area called **allowed area** or **allowed set**

$$A \in R^n \qquad (4)$$

Vector x varying in set A vector **F** elapses criterion value set Φ [6]. Multi-objective problem solving gives a set of points which complies with Pareto (Pareto 1906) optimality definition. In our case we have two objectives maximize true alerts rate and minimize false alert rate:

$$\text{Maximize } f_2(x) \qquad (5)$$
$$\text{Minimize } f_1(x) \qquad (6)$$

The parameter vector X is composed from system parameters that optimization is made by system administrator or in other words Decision Maker (DM). They can be adjusted in certain limits. So the parameter vector has the allowed area $A \in R^n$. As system parameters can vary in certain limits we can state that optimization has constraints. Our task is to eliminate the Decision Makers experience in the first stage of optimization. Just assume that DM does not have any experience in air traffic control. He is not able to evaluate the importance of system parameters and assign any weights them. Just imagine that he has a set of system parameters and adjustable limits of each of it. Sometimes this set can have hundreds of these parameters. He has to adjust them step by step and find the best set which maximizes true alert rate and minimizes false alert rate. So it's a trial for founding's and mistakes. In this case the multi-optimization methods could be used. We can simply follow the DM actions. First of all he takes the primary set of system parameters. He changes one parameter and evaluates the results. After that he changes another one parameter and again evaluates the results. In fact we can say that he has two sets of system parameters and passes the better parameter set for new evaluation. It is done until the results satisfy the DM. If we summarize his actions we will find such algorithm:

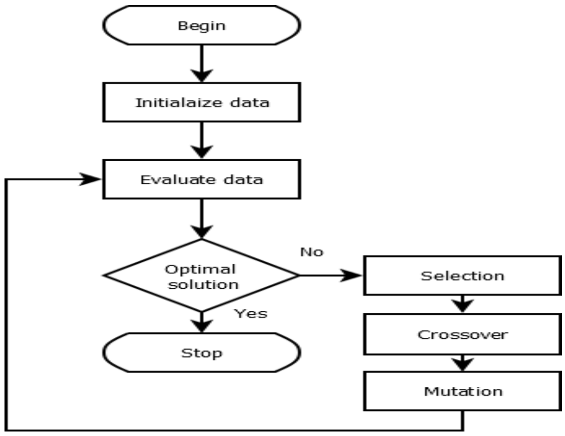

Figure 3: Simple genetic algorithm.

If we look closer we can define these stages:

Table No.2

GA stages and DM actions

GA stages	DM actions
Create initial population	initial system parameters
Compute the objective values of solution calculation	Select the acceptable false warnings rate);
Calculate the objective fitness values	Run simulation and analyze if solution is good enough
Select the fittest values for reproduction	If solution is not good enough decide which parameters could be changed
Create the individuals from mating pool by crossover and mutation	Create new parameter set
Evaluate the solution candidates	Run simulation and analyze if solution is good enough

In fact these actions are quite similar for genetic optimization algorithms. Multi-objective optimization problems are often solved using single-objective optimization methods. However genetic methods deal with multi-objective problem directly. Holland (1975) described genetic algorithms. These algorithms are quite. They are effective independently of objective function nature and constraints. These methods use random selection and information from previous iterations to improve all population. Genetic algorithms are the global optimization algorithms. They provide the solution which is Pareto optimal globally. The genetic algorithms can be grouped on the basis of methods components and the way they search for solutions. Usually two groups are set. The first group uses elite selection and the second does not [5]. Algorithms with elite selection let the best solution enter new population without any changes. The first group methods are VEGA, MOGA, NPGA, NSGA. The second group methods are NSGA-II, SPEA, SPEA 2, PAES, PESA (Guliashki et al. 2009).

2 Genetic algorithm analysis for STCA optimization

First off all we need to say a few words about Safety Nets system parameters. The extract of them is presented in picture below:

Table No.3

Example of system parameter set

System parameter	Value	Lower limit	Upper limit	Step
RDF_STCA_RATE_THRESHOLD	-500FTM	0	4000	100
STCA_LEVEL_THRESHOLD	1000.0FT	0	2000	100
RDF_STCA_LOWER_SPEED	N30	0	120	10
RDF_RAC_MAX_SPEED	990	50	990	10
RDF_STCA_LOWER_ALT	-250FT	-600	600	100
RDF_MAX_LEVEL_OVERSHOOT	200FT	0	1000	100
RDF_STCA_UNCOUPLED_TRACK_ROC	60.0FTS	10	100	10
RDF_STCA_UNCOUPLED_TRACK_ROD	60.0FTS	20	100	10
RDF_STCA_WARNING_TIME	60	30	120	10
RDF_STCA_IMMINENT_TIME:	30	0	60	10
RDF_STCA_CONFLICT_QUALITY_THRESHOLD	0.7	0,5	0,9	0,1
RDF_STCA_COUNTER_THRESHOLD	2	1	5	1
RDF_STCA_SOLVED_COUNTER_THRESHOLD	2	1	5	1
RDF_STCA_SEVERE_SEPARATION_DISTANCE	3.0NM	2	10	1
RDF_STCA_ALT_SEP_THRESHOLD_ACC	F290	100	500	100
RDF_STCA_ALT_SEP_LOW_ACC	F10	1	20	1
RDF_STCA_ALT_SEP_HIGH_ACC	F10	1	20	1
RDF_STCA_FINE_PLANAR_SEP_ACC	3.0NM	2	10	1
RDF_STCA_LOOK_AHEAD_TIME_ACC	90	60	120	10
RDF_STCA_ALT_SEP_THRESHOLD_APP	F290	100	500	100
RDF_STCA_ALT_SEP_LOW_APP	F10	1	20	1
RDF_STCA_ALT_SEP_HIGH_APP	F10	1	20	1
STCA_FINE PLANAR_SEP_APP	3.0	2	10	1
RDF_STCA_LOOK_AHEAD_TIME_APP	90	60	120	10
RDF_STCA_EXCLUDE_VFR	TRUE	TRUE	FALSE	
RDF_STCA_EXCLUDE_IFR	FALSE	TRUE	FALSE	

In fact we can say that each x_i is a real parameter subject to $a_i \le x_i \le b_i$ for some constants a_i and b_i. As it was mentioned earlier we have two objective functions maximize true positive alerts rate and minimize false alerts rate. The question is how to evaluate these objective functions. First we have to analyze the warning and decide if it is true or not. So we have to implement automated decision maker such as an expert system [3]. It could be done if we measure the distance between two aircraft in horizontal and vertical spaces. Separation minima's are known. The distances between aircraft could be compared with these minima's with added some buffer distance. The buffer distance is needed because the warning shall be produced at least two minutes in advance. So the evaluation of true and false warning criteria *C* could be expressed as follows (it should be defined in vertical and horizontal planes):

$$\mathrm{C} = sep\,min + buffer \qquad (7)$$

Now we can say that alert is true if the distance is lower or equal to criteria *C*. If the alert was produced when the distance is greater that criteria *C* the alert was false. Of course such evaluation has constraints as it depends on added buffer.

Another question is how evaluate the objective functions. As it was mentioned earlier we have two objective functions. The objective evaluation could be expressed as follows:

$$max\ (f_2(x) - f_1(x)) \qquad (8)$$

The formulation (8) is true if we use two test scenarios with In other case we can evaluate by the percentage of true alerts to false alerts. Next step is to design certain flight trajectories in order to get the alerts. They can be taken from Eurocontrol recommendations [1]. The test scenarios could be:

- Simultaneous lateral and vertical convergence.
- Aircraft converging laterally at the same flight level.
- Departure from Flight Level Situation - 1000ft.
- Departure from Flight Level Situation - 2000ft.
- Aircraft laterally slow closing on the same flight level.
- An aircraft levels off at an occupied level (optional input of CFL).
- Aircraft climbing and converging vertically.
- Aircraft proceeds out of exclusion region into imminent conflict.
- One aircraft turning towards another into conflict
- Two aircraft turning towards each other into conflict
- Aircraft cross laterally then cross vertically.

However these scenarios give true alerts. So we need to generate two types of scenarios: one which gives true separation loss and second one which does not make separation infringement. These scenarios should be very similar. In other case we could use the real recorded data.

Another step is to set the initial population. In fact it is the set of system parameters which certain values and can be changed at certain steps:

$$x_{i+1} = x_i + \alpha \qquad (9)$$

Where

x_i – starting parameter value,
α – parameter step,
x_{i+1} – next parameter value.

So the initial population is set. So we have initial population composed of system parameters representing criterion vector *x*. Next step is to divide initial set into two parent's subsets. Normally the criterion vector is split into two parts which represents two parent chromosomes. The split point is called crossover point. It could be done simply splitting the initial set into two parent sets. That means we use one point crossover. The usual method of applying genetic algorithms to real-parameter problems is to encode each parameter as a bit string using either a standard binary coding. However they could be applied to real parameter spaces whose chromosomes are vectors of floating point numbers and whose alleles are real numbers [2]. In this case, one child gets some of its parameters from one parent, and some of its parameters from the other parent. We can assume that parents are X and Y, corresponding to parameter vectors [2]:

$$x = x_1, x_2, \dots, x_m \qquad (10)$$

and

$$y = y_1, y_2, \dots, y_m \qquad (11)$$

So one child has the parameter vector $(x_1, x_2, \dots, x_i, y_{i+1}, \dots, y_m)$ and second one $(y_1, y_2, \dots, y, x_{i+1}, \dots, x_m)$. The new child chromosomes are composed from the parents. How we have to decide which parameter should be mutated. The mutation could be done in positive direction. We have to decide at what probability each parameter should be mutated in order to run through all possible values or in other words run through all parameter space. We can use the random selection

and one parameter for one child. The mutation is done according (9). After mutation we can run our test iterations and evaluate the objective function (8).
Next step is to decide which parameter set should be passed to the next generation. If one parameter gives better results it should be passed to the next generation. Iterations could be continued until the whole allowed area has been searched.

3 Primary scheme for experiments

In previous chapters we discussed about similarity simple genetic optimization algorithm and decision maker's actions, touched the possibility of GA application for real parameter systems. Now we can say what is needed for the GA testing system:

- Software for GA application;
- Flight trajectories simulation;
- Air traffic control simulator;
- STCA warning registration data;
- Software for objective function fitness evaluation.

Summarizing, we can form such testing platform:

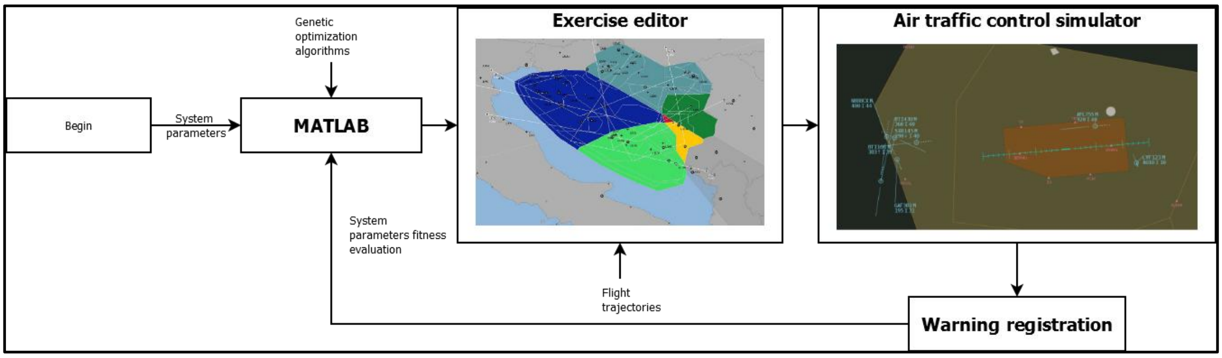

Figure 4: Testing platform.

MATLAB software does selection, crossover, mutation and evaluation [4]. Exercise editor forms flight trajectories for certain test iterations. Scenarios and particular system parameter sets are run air traffic control simulator. Optimal parameter set can be tested using recorded flight data.

4 Conclusion

The problem of true and false Safety Nets warnings was analyzed. The false warning rate could be minimized by adjusting system parameters. The genetic optimization algorithms can be used in this area. The main difference is that real parameters shall be used instead of binary coding. The testing platform configuration was suggested for experiments.

Acknowledgment

The first author gratefully acknowledges support of SE „Oro navigacija" – Air Navigation Service Provider of Lithuania .

References

[1] EUROCONTROL Guidance Material for Short Term Conflict Alert Appendix A: Reference STCA System Edition: 2.0 Edition Date: 19 May 2009.

[2] A. H. Wright, “Genetic Algorithms for Real Parameter Optimization”, Department of Computer Science University of Montana Missoula, Montana 59812. 1991.

[3] Carlos M. Fonseca and Peter J. Fleming “Multiobjective Optimization: Formulation, Discussion and Generalization”. Dept. Automatic Control and Systems Eng. University of Sheffield. Sheffield S1 4DU, U.K. 1993.

[4] Andrew Chipperfield, Peter Fleming, Hartmut Pohlheim, Carlos Fonseca “Genetic Algorithm for use with MatLab”. Dept. Automatic Control and Systems Eng. University of Sheffield. Version 1.2, User’s Guide. 1994.

[5] Ernestas Filatovas. Daugiakriterinių optimizavimo uždavinių sprendimas ineraktyviuoju būdu. Daktaro disertacija Technologijos mokslai, informatikos inžinerija (07T) Vilnius, 2012. (in Lithuanian)

[6] A. Zilinskas. Matematinis Programavimas Vilnius 2005. (in Lithuanian).

[7] EUROCONTROL Guidance Material for Short Term Conflict Alert Ed. 2.0, May 2009.