COMPARISON AND IMPROVEMENT OF METHODS TO SOLVING THE PROBLEM OF OPTIMAL FLIGHT ROUTING

Lolita A. Mirzoyan
Moscow Aviation Institute (state technical university)

Keywords: *routing, neural networks, genetic algorithm, risk function, Halton method*

Abstract

The review, analysis and comparison of some approaches to solving a problem of aircraft flight routing were carried out in activity. The new methods of the problem solving based on improving of existing methods and on their effective combination are offered also.

The problem is solved by an example of horizontal flight routing for specified in any moment coordinates of initial, final and intermediate destinations. The minimum of aircraft range was considered as a main optimization criterion of the routing.

1 Introduction

Today the problem of civil and military aircraft routing is yet actual. 40-60 kg of fuel is consumed for each kilometer of flight range. That is why the rational planning of flight will significantly reduces flight cost. Besides, it is important to know not only how to plan route before flight, but also how to make re-planning in real time.

In the first part of activity existing methods of routing are considered, their analysis is carried out, and results of their implementation are represented.

The second part of activity is devoted to new developed and researched methods.

The problem is solved by an example of horizontal flight routing for specified two-dimensional coordinates of initial, final and intermediate destinations. Notice that the dimensionality of coordinates (X,Y) does not lower the quality of the solution, because for the most offered methods taking into account altitude (third coordinate Z) has an effect only on calculation of distances between destinations.

The problem is to find the flight route from initial to final destination with flying through all intermediate one's. The minimum of aircraft range was considered as a main optimization criterion of the routing.

2 Well-known methods. The review, results of the application, comparison.

2.1 Branch-and-bound algorithm

Branch-and-bound algorithm is one way of exhaustive search organization improving it by series cut of sets routes that have not been optimal.

An input data to solve the problem using this algorithm is the matrix of distances between destinations. The example of such a matrix for 5 destinations is shown in fig.1, where d_{ij} is the distance between i and j destinations, ∞ is the symbol of infinity, ensuring impossibility of flight within one destination. This matrix can be asymmetrical because of taking into account altitude, external factors.

from \ to	1	2	3	4	5
1	∞	d_{12}	d_{13}	d_{14}	d_{15}
2	d_{21}	∞	d_{23}	d_{24}	d_{25}
3	d_{31}	d_{32}	∞	d_{34}	d_{35}
4	d_{41}	d_{42}	d_{43}	∞	d_{45}
5	d_{51}	d_{52}	d_{53}	d_{54}	∞

Fig. 1. The Matrix of Distances for Branch-and-Bound Algorithm

This algorithm is conventionally applied to solve the so-called traveling salesmen problem, in which required route is closed. For considered problem the route should be unclosed at fixed initial and final destinations. According to this condition it is offered to use a modified matrix of distances. The example of modified matrix for 5 destinations is shown in fig.2., where 1 destination is initial, 5 is final.

from \ to	1	2	3	4	5
1	∞	d_{12}	d_{13}	d_{14}	d_{15}
2	∞	∞	d_{23}	d_{24}	d_{25}
3	∞	d_{32}	∞	d_{34}	d_{35}
4	∞	d_{42}	d_{43}	∞	d_{45}
5	∞	∞	∞	∞	∞

Fig. 2. The Modified Matrix of Distances for Branch-and-Bound Algorithm

The branch-and-bound algorithm can be presented as the following sequence of steps:

1. The set of feasible routes is partitioned into two subsets: including and excluding selected flight step.
2. The lower estimation of each subsets is made as a minimum of route length; subset, which is marked by the less estimation is taken as a winner and define whether selected flight step will be included in final route or not.
3. The incorrect flight sets (broken the problem logic, for example, closed the route) are excluded from subset-winner.
4. If the subset-winner consist of only one element, the algorithm is finished, otherwise the subset-winner is returned to the first step as the set of feasible routes.

The solution search tree is illustrated in fig.3.

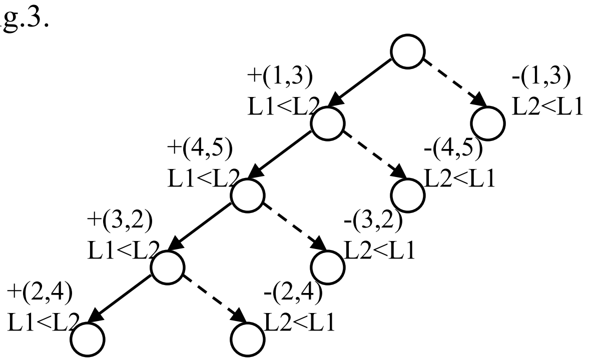

Fig. 3. The Solution Search Tree for Branch-And-Bound Algorithm

The results of using Branch-and-bound algorithm are presented in fig. 9.

Thus, the branch-and-bound algorithm allows to find optimal (in sometimes near-optimal) route. A disadvantage of the algorithm lies in its low response: the time it takes for solution obtaining increases exponentially with increasing number of destinations. Because of this, the algorithm is of limited usefulness in real time and in some cases at a lot of destinations the algorithm can not be used as well as the exhaustive search.

2.2 The method of search of a risk function analytical form represented as polynomial

The risk function permitting to determine the following more expedient and prime destination in each point of fly-around was considered as the elementary analytical form of decision-making. The prime was the destination with the minimal risk function.

The polynomial of the second degree with respect to increments of geographical coordinates X and Y of *j* destination location on a map was taken as such a risk function:

$$F_j = b_1 \cdot X_j + b_2 \cdot Y_j + b_3 \cdot X_j^2 + b_4 \cdot Y_j^2 + b_5 \cdot X_j \cdot Y_j,$$

where X_j, Y_j are differences between the coordinates of *j* destination and the coordinates of the current destination.

The coefficients $b_1 \ldots b_5$ were defined during the self-learning process.

The algorithm of self-training for polynomial coefficients definition consist of the following steps:

1. some reference value of vector $b_1(0)$, $b_2(0)$, $b_3(0)$, $b_4(0)$, $b_5(0)$ is set ;
2. routing of flight in this untrained condition is carried out; the frequency of making correct solutions is determined;
3. deflections of coefficients b_i + db are created in turn; the success of a behaviour is estimated again;
4. the best value is taken as a new reference value for the next stage of self-training.

In this way with new elements of neural computer science on the basis of self-training using 30 examples each consisted of 10 destinations the answer was obtained:

$$\mathrm{F_j} = 0.5 \cdot X_j + 0.4 \cdot X_j^{\,2} + 0.1 \cdot Y_j^{\,2}.$$

Such a formula will be quite agreed with the sense of the problem posed.

2.3 The genetic algorithm

The genetic algorithm is a method, which combines exhaustive search and local-gradient methods. A disadvantage of exhaustive search lies in its long-duration implementation. A disadvantage of a gradient method lies in its convergence to local minimum of objective function. The genetic algorithm allows to overcome disadvantages and to use to the full advantages of both methods. The crossover and mutation operators realize the exhaustive part of the method and the best solution selection operators realize the gradient search part of the method.

The feasible routes are considered as individuals. Route parameters are written as one chromosome i.e. the vector of length *n* (where *n* is number of destinations). The first gene always contains an order number of initial destination and last gene contains an order number of final destination. An example of chromosome for *n*=20 are presented on fig. 4.

1	10	4	8	18	19	5	11	17	2	13	9	6	3	12	7	14	16	15	20

Fig. 4. The Chromosome for Genetic Algorithm

Such chromosomes are called as “enumerable chromosome with unique genes”.

The traditional mutation operators aren’t used for such type of chromosomes, since it will cause the incorrect route. The modified mutation is proposed as swop of two randomly selected genes.

The traditional crossover operators aren’t used for such type of chromosomes is also incorrect, therefore more complex scheme of two-point crossover is offered to use.

2.4 The Hopfield neural network

A continuous Hopfield neural network was used to solve the problem. This network consisting of n^2 continuous neurons (n is the number of destinations), arranged in the form of an n×n matrix. The *i* row in the matrix corresponds to the *i* destination, the *i* column in the matrix corresponds to the *i* order in route. An example of network is shown in fig. 5.

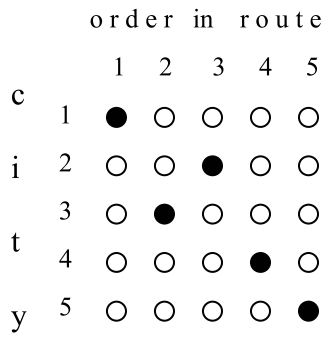

Fig. 5. The Matrix for Hopfield Neural Network

The dynamics of this network can be characterized by an equation:

$$\frac{\mathrm{dU_{ij}}}{\mathrm{dt}} = -\frac{\mathrm{U_{ij}}}{\tau_{\mathrm{ij}}} + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\mu=1}^{n}\sum_{\nu=1}^{n} W_{ij\mu\nu} \cdot V_{\mu\nu} - I_{ij},$$

where $U_{ij}(t)$ is an internal state of neuron; τ_{ij} is the value of the time constant for neurons; $W_{ij\mu\nu}$ are the weights of the neurons; V_{ij} is the external state of neuron; I_{ij} is the value of external threshold. The external state of neurons is continuous within the range [0,1] and can be defined in accord with an activation function:

$$\mathrm{V_{ij}} = g(U_{ij}) = \frac{1}{1 + e^{-\lambda U_{ij}}},$$

where λ is the gain parameter of the sigmoid function. There are 4 constraints of the problem, which should be satisfied to obtain the feasible and “good” solution:

- each destination can be visited only once;
- only one destination can be visited at a time;
- the final route is to include all n destinations;
- the total aircraft range is to be the minimal;

A proper energy function is defined as

$$E = \frac{A}{2}\cdot\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\nu\neq j}^{n} V_{ij}\cdot V_{i\nu} + \frac{B}{2}\cdot\sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\mu=1}^{n} V_{ij}\cdot V_{\mu j} +$$

$$+\frac{C}{2}\cdot\left[\sum_{i=1}^{n}\sum_{j=1}^{n} V_{ij} - n\right]^2 +$$

$$+\frac{F}{2}\sum_{i=1}^{n}\sum_{\mu=1}^{n}\sum_{j=1}^{n} d_{i\mu}V_{ij}\cdot(V_{\mu,j+1}+V_{\mu,j-1}) =$$

$$= E_1 + E_2 + E_3 + E_4,$$

where d_{ij} is the distance between i and j destinations, A, B, C, D are penalty parameters, that are chosen to reflect the relative importance of each term (E_1, E_2, E_3, E_4) in the energy function.

The analysis of the above formulas allows to define the main parameters of neural network:

- the weights:

$$W_{ij\mu\nu} = -A\cdot\delta_{i\mu} - B\cdot\delta_{j\nu} + C\cdot\delta_{i\mu}\cdot\delta_{j\nu}$$
$$- D\cdot d_{i\mu}\cdot(\delta_{\mu,j+1}+\delta_{\mu,j-1}),$$

where δ_{ij} is the Kronecker delta and is defined as

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{for } i \neq j \end{cases};$$

- the external threshold:

$$I_{ij} = -\frac{C}{2} + (A+B).$$

The penalty parameters were offered by J. Hopfield [1] as constants A=B=500, C=200, D=500, τ=1, λ=50.

The results of using continuous Hopfield neural network are presented in fig. 9.

Thus, a disadvantage of the Hopfield neural network lies in high probability of obtaining a local minimum of aircraft range or incorrect route. The advantages of this network lies in its fast response at hardware realization. It is necessary to take into account, that every so often it is required to find the route in real time and the main factor is the response of algorithm instead of global optimum definition. Therefore it is possible to consider the Hopfield neural network as effective method. Moreover, there is an analogous electronic circuit that uses nonlinear amplifiers and resistors for realizing continuous Hopfield neural network which can rapidly solve such difficulty problem.

2.5 The Kohonen neural network

It is considered the self-organizing Kohonen neural network that can be trained to detect groups (clusters) of input vectors.

There are to types of Kohonen neural networks: so called Kohonen layers with unordered neurons and the Kohonen maps with ordered neurons. Our problem is solved by using one-dimensional Kohonen map, that has a necessary property to indicate the data structure in such way to near located neurons correspond to the near located clusters of input vectors. The architecture of Kohonen neural network is very simple: 2 input neurons, to which coordinates of destinations (X and Y) are consequently fed ; n output neurons (fig.5).

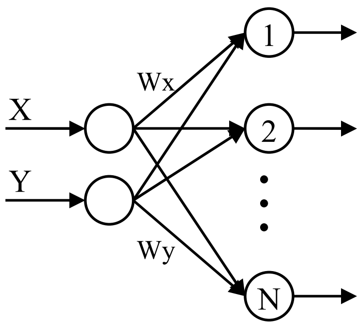

Fig. 6 The Kohonen Neural Network Architecture

Coordinates of all destinations generate a learning set for the network.

The algorithm of Kohonen neural network learning can be presented as following sequence of steps:

1. The weights Wx and Wy are initialized with random values.
2. One of destinations is selected at random, its coordinates are fed to input neurons.
3. The Euclidean distance between coordinates of selected destination and weights is defined for each output neutron:

$$D_j = (x - w_{xj})^2 + (y - w_{yj})^2, \quad j = 1,..,n.$$

4. The neuron-winner is determined as providing least Euclidean distance:

$$D_{win} = \min_j\{D_j\},$$

where *win* is a number of neuron-winner.

5. Weights of network are modified according to radius of attraction *r(j,win)*:

$$w_{xj}(t+1) = w_{xj}(t) + r(t)\cdot\alpha(t)\cdot(x - w_{xj});$$
$$w_{yj}(t+1) = w_{yj}(t) + r(t)\cdot\alpha(t)\cdot(y - w_{yj}),\quad j = 1,..,n,$$

where *α(t)* is a learning rate parameter, which is decreased in time as following:

$$\alpha(t) = \frac{1}{t}.$$

6. Return to 2 step and repeat above procedure for all destinations.
7. The values of *r(t)* and *α(t)*.are decreased.

This procedure is continued until weights cease to change.

After the end of learning process the destination location in final route is determined by location of associated output neuron in the network output layer.

The results of using Kohonen neural network are presented in fig. 9.

As often happens, two or more destinations associate with one output neuron. This situation can be interpreted as follow: the local ordering of these destinations is of no concern and it is required only local optimization of the route segment. At few ten of destinations such optimization can correct the route length up to 25% (see 3.5, 3.6). At few hundred of destinations this optimization as a rule does not improve the results, and requires just additional computational time, that is why it does not used now.

The capability of using in real time without accuracy loss and computational time increasing can be lied to the advantageous of this network. Really, if new destination is added into the route it is enough to feed its coordinates on the input of neural network, then to define a compliance with corresponding cluster and to continue flight with according to searched route. If distance between clusters and new destination is very long it has meaning to conduct a recalculation, but weight coefficient initiate not by random value but by value obtained previously during leaning. It increases speed of decision making in several times.

3. New methods and offering for known methods improvement

Following methods of the problem are offered, developed and studied by me:

- Halton method for choice of coefficients in polynomial which is analytical form of risk function;
- Halton method for choice of coefficients in energy function of Hopfield neural network;
- Repeatedly use of Kohonen neural network;
- Use of Kohonen neural network in combination with other methods.

3.1 The Halton method. Brief description.

The Halton method is a little-known method of global multiparametric optimization and constitutes a deterministic analog of global random search. The method consists in use of so called pseudo-random uniformly sequences of points as the trial test points.

During random search independent random points are selected as trial test points distributed uniformly in feasible range. The probability of a point hitting in the small vicinity of minimum U point is equal $P=1-(1-U)^N$ and tend to 1 at $N\to\infty$, i.e. the method is convergent.

During nonrandom search points are uniformly selected in feasible range (fig. 7) .

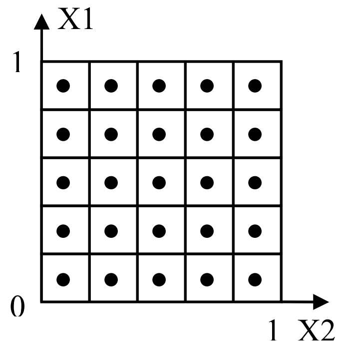

Fig. 7. The Trial Test Points For Uniformly Nonrandom Search

However this principle often is not effective. If the objective function $f(x_1,x_2\ldots x_n)$ essentially depends from m < n arguments, then $N\text{-}M^m$ (M is the number of various values arguments) will be considered needlessly. So if n=4, m=2 and M=10 then number of such values will be equal $N-M^m = M^n- M^m = 10^4-10^2 = 9900$ from a total number $N = 10000$ calculated values. The random grids are deprived of this disadvantage, since the probability of identical abscissas appearance at several random points is equal to zero. However inspection and control of random search is impossible, it means that each time the iterations number will be various . It is expedient to assign trial test points intelligently (fig. 8).

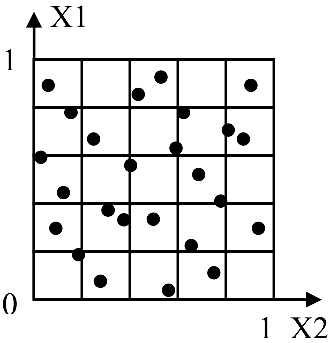

Fig. 8. The Trial Test Points for Halton method

As is seen in fig. 7, calculation values of the function in points of such grid will give N various values that provides a better guide to the function and its range.

Idea of Halton method consists in use of Halton sequence. In [2] is proved, that the global search on any points from pseudo-random uniformly sequences is convergent at rather large N.

The Halton sequence. If $r_1,r_2,\ldots,r_n$ are the prime numbers, then the Halton sequence is the sequence of points with Cartesian coordinates $P_i=(p_{r1}(i), p_{r2}(i),\ldots, p_{rn}(i))$, $i=1,2,\ldots$, where $p_r(i)$ is the numerical sequence defined below.

If in r-based notation $i=a_m a_{m-1}\ldots a_2 a_1$, then in r-based notation $p_r(i)=0.a_m a_{m-1}\ldots a_2 a_1$ (a_s are the integer r-based notation digits, i.e. they are equal one of the follow values $0,1,\ldots, r-1$). So in a decimal notation is obtained

$$i = \sum_{s=1}^{m} a_s \cdot r^{s-1}; \; p_r(i) = \sum_{s=1}^{m} a_s \cdot r^{-s}.$$

For example, first ten values are presented in table 1.

i	1	2	3	4	5	6	7	8	9	10
$p_2(i)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{5}{8}$	$\frac{3}{8}$	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{9}{16}$	$\frac{5}{16}$
$p_3(i)$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{9}$	$\frac{4}{9}$	$\frac{7}{9}$	$\frac{2}{9}$	$\frac{5}{9}$	$\frac{8}{9}$	$\frac{1}{27}$	$\frac{10}{27}$
$p_5(i)$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$	$\frac{1}{25}$	$\frac{6}{25}$	$\frac{11}{25}$	$\frac{16}{25}$	$\frac{21}{25}$	$\frac{2}{25}$
$p_7(i)$	$\frac{1}{7}$	$\frac{2}{7}$	$\frac{3}{7}$	$\frac{4}{7}$	$\frac{5}{7}$	$\frac{6}{7}$	$\frac{1}{49}$	$\frac{8}{49}$	$\frac{15}{49}$	$\frac{22}{49}$
$p_{11}(i)$	$\frac{1}{11}$	$\frac{2}{11}$	$\frac{3}{11}$	$\frac{4}{11}$	$\frac{5}{11}$	$\frac{6}{11}$	$\frac{7}{11}$	$\frac{8}{11}$	$\frac{9}{11}$	$\frac{10}{11}$

Table 1. The Halton Sequence for 10 Values

The advantages of method are:

- more fast convergence in comparison with random search at high;
- calculation of Halton sequence is rather simple and easily to programming;
- For application of this method it is not required not only differentiability of objective function (this condition is necessary for application of gradient methods), but even its analytical form - needs only to have an opportunity to calculate values of the function in any points of its range;
- In difference from other methods this method is effective for solving the problem of large dimensionality (n> 50).

3.2 Halton method for search of analytical form of risk function

Notice, that method for search of coefficients in polynomial which is analytical form of risk function, is analog of gradient method with its disadvantages. It is proposed to use of Halton method for search of coefficients in polynomial. It is considered the learning on 30 examples for 10 destinations.

As the input data 30 examples of optimal routes consisted from 10 destinations was used.

The algorithm of Halton method for polynomial coefficients definition consist of the following steps:

1. Using of Halton method the next vector of polynomial coefficients is created (in range [0,1]).
2. The value of polynomial is calculated for each of examples, the number of predicted next

destination is defined from minimum polynomial point of view.
3. The number of predicted next destination from each example is compared with corresponding one from optimal route. Counter of agreements is increase on 1.
4. If given number of considered vectors is reached the process is finished and it is conducted the obtained results analysis. Else it is conducted return back to step 1.

The results of using Halton method for search of analytical form of risk function are presented in fig. 9.

3.3 Halton method in Hopfield neural network

As stated above (see 2.4), the energy function has many local minima. The reason is that it comprises several terms, each of which is competing to be minimized.

An infeasible solution to the problem will arise when at least one of the constraint penalty terms is non-zero. The solution is “good” but not feasible.

Alternatively, all constraints may be satisfied, but a local minimum may be encountered that does not globally minimize the objective function, in which case the solution is feasible, but not “good”.

The solution to this trade-off problem is to find the optimum values of the penalty parameters that balance the term of energy function and ensure that each term is minimized with equal priority. Only than will the constraint terms be zero (a feasible solution) and the objective function be also minimized (a “good” solution).

It is proposed to use the Halton method for penalty parameters search.

The main idea of such approach consist in:

- the calculating of the objective function value (route length) in each trial test point, generated by Halton sequence, is conducted with dynamic Hopfield neural network under appropriate weights;
- The solution is consistent with the best objective function value

The realization of this method demonstrated that such modified Hopfield network is more precisely and faster as its original one.

3.4 The repeatedly use of Kohonen neural network

As noted above (see 2.5), every so often two or more destinations associate with one output neuron of learned Kohonen neural network. This network’s ability of destinations classification and grouping is taking advantage of as preliminary stage to solve the problem.

The algorithm of repeatedly use of Kohonen neural network can be presented as following two consequently steps:
1. The algorithm of Kohonen neural network learning (see 2.5) is executed for the destinations grouping and obtaining the route between centers of groups.
2. The algorithm of Kohonen neural network learning is executed again for obtaining the route between destinations inside each group. For each group the destination nearest to the previous in route (which is obtained on the 1 step) center of group is determined and is considered as the initial destination. The destination nearest to the next in route center of group is determined and is considered as the final destination.

3.5 Use of Kohonen neural network in combination with other methods

This method is realized by using exhaustive search or Hopfield neural network on the 2 step of algorithm considered above (see 3.4).

The results of using Kohonen neural network in combination with exhaustive search are presented in fig. 9.

4 The comparative analysis of well-known and new methods implementation

The experiments with well-known methods suggest that

- The branch-and-bound algorithm and the Hopfield neural network may be better from smallest problem size. (less than 50 destinations).

- The best solution for small problem size (50 -100 destinations) can be obtained by Kohonen neural network .
- The genetic algorithm is more effective on larger problem size (more than 100 destinations).

The experiments with new methods suggest that

- The modified Hopfield neural network and may be better from small problem size. (50-100 destinations).
- The Kohonen neural network in combination with exhaustive search is the best on large problem size (more than 100 destinations).
- If there is the time for example generation on large problem size the method based on search the analytical form of risk function will always effective.

5 Conclusions

Analysis and comparison of existing methods to solving a problem of aircraft flight routing are shown:

1. A disadvantage of the Branch-and-bound algorithm lies in its low response: the time it takes for solution obtaining increases exponentially with increasing number of destinations. A disadvantage of the Hopfield neural network lies in high probability of obtaining a local minimum of aircraft range or incorrect route. The advantages of this network lies in its fast response at hardware realization.
2. The branch-and-bound algorithm and the Hopfield neural network may be better from smallest problem size (less than 50 destinations).
3. Kohonen neural network is the best solution for small problem size (50 -100 destinations).
4. The genetic algorithm is more effective on larger problem size (more than 100 destinations).
5. Modified Hopfield network is more precisely and faster as its original one and may be better from small problem size (50-100 destinations).
6. The Kohonen neural network in combination with exhaustive search is the best on large problem size (more than 100 destinations).
7. If there is the time for example generation on large problem size the method based on search the analytical form of risk function will always effective.
8. Investigation should be conducted in direct of improving the methods by effective combination those under consideration.

References

[1] Hopfield J, Tank D. Neural computation of decisions in optimization problems. *Biological cybernetics*, Vol. 52, pp 141-152, 1985.

[2] Medynskiy M, Antoniy E**.** The numerical methods of nonlinear optimization: algorithms and programs. MAI, 2003 (in Russian).

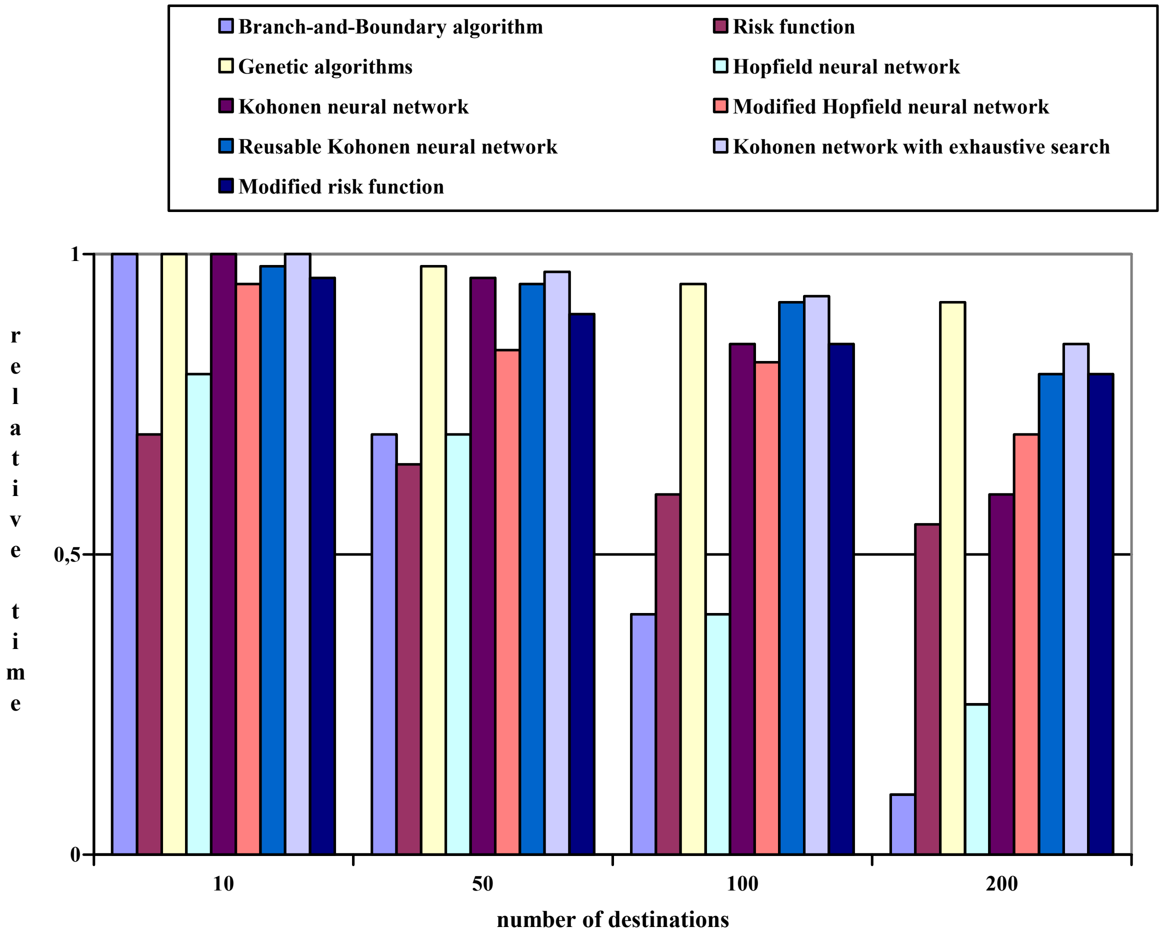

Fig. 9. The results of Using Well Known and New Methods under Consideration