

Evaluation of nature inspired metaheuristics for search and reconnaissance operations by rotary-wing aircrafts**Avaliação de meta heurísticas inspiradas na natureza para operações de busca e reconhecimento por aeronaves de asa móvel**

DOI:10.34117/bjdv5n8-044

Recebimento dos originais: 14/07/2019

Aceitação para publicação: 19/08/2019

André M. Yokoyama

Mestre em Modelagem Computacional pelo Laboratório Nacional de Computação Científica
Instituição: Laboratório Nacional de Computação Científica (LNCC)
Endereço: Av. Getúlio Vargas, 333 – Quitandinha, Petrópolis – RJ, Brasil
E-mail: andremy@lncc.br

Antonio R. Mury

Doutor em Engenharia de Produção pela Universidade Federal do Rio de Janeiro
Instituição: Laboratório Nacional de Computação Científica (LNCC)
Endereço: Av. Getúlio Vargas, 333 – Quitandinha, Petrópolis – RJ, Brasil
E-mail: a.roberto.m@gmail.com

Bruno Schulze

Doutor em Ciência da Computação pela Universidade Estadual de Campinas
Instituição: Laboratório Nacional de Computação Científica (LNCC)
Endereço: Av. Getúlio Vargas, 333 – Quitandinha, Petrópolis – RJ, Brasil
E-mail: schulze@lncc.br

Mariza Ferro

Doutora em Modelagem Computacional pelo Laboratório Nacional de Computação Científica
Instituição: Laboratório Nacional de Computação Científica (LNCC)
Endereço: Av. Getúlio Vargas, 333 – Quitandinha, Petrópolis – RJ, Brasil
E-mail: mariza@lncc.br

ABSTRACT

The main objective of this work is the evaluation of two nature in-spired meta-heuristics, Genetic Algorithms and Ant Colony, for the development of an application that can generate optimized routes for aircraft, attending the requirements of the Brazilian Navy. This work presents the methods developed, complying with two main constraints: checkpoints mobility and limited aircraft autonomy. It also presents the results of tests performed with the methods developed and an evaluation of their performances.

Keywords: Genetic Algorithms, Ant Colony, Metaheuristic.

RESUMO

O objetivo principal deste trabalho é a avaliação de duas meta-heurísticas espinhais da natureza, Algoritmos Genéticos e Colônia de Formigas, para o desenvolvimento de uma aplicação que possa gerar rotas otimizadas para aeronaves, atendendo às exigências da Marinha do Brasil. Este trabalho apresenta os métodos desenvolvidos, obedecendo a duas restrições principais: a mobilidade dos pontos de checagem e a autonomia limitada das aeronaves. Apresenta também os resultados dos testes realizados com os métodos desenvolvidos e uma avaliação de seus desempenhos.

Palavras-chave: Algoritmos Genéticos, Colônia de Formigas, Metaheurística

1. INTRODUCTION

In Brazil the sum of the areas of the Exclusive Economic Zone and Continental Shelf is nearly 4.5 million square kilometers. This territory contains our reserves of oil and natural gas and all the mineral wealth of the subsoil [Wiesebron 2013]. However, to exercise its rights, Brazil must be able to exercise its maritime power. This work deals with the problem of determining the best way to perform reconnaissance operations carried out by naval force operating in a maritime area. The reconnaissance operations consist in the search of useful information in a given area in order to guide the use of military means. These operations are carried out by reconnaissance and attack helicopters carried on board Navy vessels. The goal is to determine the best route that this rotary-wing aircraft will do, in order to be able to identify the largest number of targets in a given area.

This problem is similar to the Traveling Salesman Problem (TSP), but has significant differences from the classical model, as both the point of origin and the destination points are mobile. Also, the “salesman” (in this case the helicopter) has a limit of distance to travel and the points to be visited (the targets) can have different priority values.

1.1. RELATED WORKS

The use of meta-heuristics inspired by nature, in particular Genetic Algorithms (GA) and Ant Colony Optimization (ACO), to solve NP-hard problems, like the TSP, have shown good results [Yang et al. 2008, Chatterjee et al. 1996]. For this reason, we chose to evaluate the use of these two meta-heuristics, GA and ACO, since the problem we are dealing with is similar to the TSP. After an extensive review of the literature were found works that make adaptations of these methods for the TSP problem, like [Li et al. 2008] for ACO and [Hassan et al. 2013] for AG. However, no studies were found that adapt these methods to the problem addressed in this work.

Based on the problem described, we developed some methods based on GA and ACO to solve the problem of air reconnaissance of targets. This work proposes the evaluation of these metaheuristics to determine the feasibility of its use to solve the problem addressed, given the constraints for the problem. The methods developed, its evaluation and results obtained are presented in Sections 3 and 4.

2. METHODOLOGY AND PROPOSAL

This section presents the methodology applied in the experiments, the proposed functions for each method and the description of the experiments. The methodology adopted in this work is divided into two main steps: The first step focuses on the development of methods based in the GA and ACO for the solution of the proposed problem, taking into account only targets mobility. The second step adapts the application created in step 1 to include a new restriction: the aircraft autonomy limit. This requires the inclusion of a new objective to the problem, the maximization of the number of identified targets, since, given the limited range of the aircraft, it may not be possible to identify all the targets.

For each step is proposed a set of tests which have characteristics in common and also specific parameters, both for GA and ACO. Common for all we have that for each test will be used two different datasets that we named 'C1' and 'C2'. These datasets are randomly generated, but following certain constraints to make them as close to a real-life scenario as possible. They include setting the targets at distances greater than 15 nautical miles from the mother ship, which is the average distance where the targets could be visually identified by mother ship itself, and with a maximum possible speed of 32 knots, equivalent to the average attack speed of military ships which is 60% than the cruise speed of commercial vessels. The coordinate system used was the Cartesian plane but the application can use the spherical or geographical coordinate if needed. The number of targets for each dataset was defined as 20.

The specific ones for GA and ACO are: based on the literature and the initial tests with the classical TSP, it was concluded that a population size for GA of 100 individuals would be the ideal for a series of 20 targets. For ACO it was defined to be equal to the number of targets in the datasets (20). Regarding the Number of individuals to be preserved or elitism for GA, we choose to use elitism to preserve the best individual of the newly generated population and to preserve the two best individuals in each generation. For GA,

the crossover probability = 80% and Mutation probability = 1%. Maximum number of generations: for GA it was evaluated the evolution curve [Yokoyama 2016] of the quality of the solutions in relation to the number of generations and it was observed any improvement in the results (in 99% cases) beyond 1000 generations. For ACO = 200.

3. PROPOSED METHODS AND RESULTS - STEP 1

This first step is focused on the development of methods based on the ACO and GA meta-heuristics. The functions for each method were developed based on the classical functions. But, to attempt the requirements of the problem addressed in this work, we propose our own methods. We are presenting the more important functions implemented, because of the limitations of space. The complete set of functions are in [Yokoyama 2016].

In this step is considered only the mobility of points (targets) to be identified. For the classic TSP the distances between points to be visited is previously known or is calculated in application startup. These distances are stored in a matrix of distances. However, due to the fact that in the problem addressed the targets are moving, the distance matrix cannot be determined at the beginning of applications. The distance between each target varies according to their order on the route as well as the order of the preceding targets. The proposed methods and results obtained, applied on the two datasets C1 and C2, are presented next.

3.1 FUNCTIONS PROPOSED FOR THE ANT COLONY OPTIMIZATION

The Exploration Function is responsible for building the route of each ant. In our proposal, the assembly of the route is always initiated by the target 0 (the mother ship). The selection of the next steps is performed by a simple roulette method where the probability of a target to be selected is directly proportional to the amount of pheromone in the edge between the current target and the candidate target and inversely proportional to the distance between them:

$$P_j = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \delta_{ij}^\beta}{\sum_{k \in N} \tau_{ik}^\alpha \cdot \delta_{ik}^\beta} & \text{if } j \in N \\ 0 & \text{if } j \notin N \end{cases} \quad (1)$$

where i is the current and j the candidate target; τ_{ij} the amount of pheromone between i and j ; $\delta_{ij} = 1/D_{ij}$ and D_{ij} the distance between i and j ; N is the set of all targets that have not been selected yet; α and β are weights defining the importance given to each

parcel. This function is also responsible for invoking our Pheromone Deposit Function and the Pheromone Evaporation Function functions [Yokoyama 2016]. For Tracks Initialization Function, all edges are started with the same amount of pheromone ($\tau_{ij} = 0.1$). Tests performed at this step using this functions are called ACO.

Functions Proposed for the Optimization Based on Genetic Algorithm

We implemented two functions for Individuals Generation to generate the initial population. The first, Individuals Generation with equal probability of selection of targets, generates a random permutation of available targets, without any bias. It is always initiated by the target 0 (the mother ship). Tests performed at this step that used this function are called GA. The second is the Individuals generation with a probability of choice of target proportional to the distance. The creation of the chromosome is always initiated by the target 0. The selection of the next alleles is performed by a simple roulette method, where the probability of a target to be chosen is inversely proportional to the distance between the current target and the candidate target given by the Equation 2.

$$P_j = \begin{cases} \frac{\delta_{ij}^2}{\sum_{k \in N} \delta_{ik}^2} & \text{if } j \in N \\ 0 & \text{if } j \notin N \end{cases} \quad (2)$$

where i is the current and j the candidate target; τ_{ij} the amount of pheromone between i and j ; $\delta_{ij} = 1/D_{ij}$ and D_{ij} the distance between i and j ; N is the set of all targets that have not been selected yet. The tests performed using this function were called GA-2.

The Mutation function implemented for this stage randomly selects any two different alleles of the individual's chromosome with the exception of the first allele of the sequence, and swaps their respective positions.

The fitness function evaluates the fitness of each individual of the population. For this step the fitness is given by a simple equation (Equation 3), since this step focus only on the impact of the mobility of the targets.

$$f_i = d_{\min} + d_{\max}d_i \quad (3)$$

where d_{\min} is the shortest distance observed in the assessed population, d_{\max} a greater distance observed and d_i the distance of the i -th individual.

Functions for Crossover and some other functions not presented here are detailed in [Yokoyama 2016]

Results for datasets C1 and C2: In Table 1 are related to the lowest and highest error obtained by each method (GA, GA-2 and ACO), the average error obtained in the tests and the percentage of results above the tolerance of 20% of error, for datasets C1 and C2. The best values are shown in bold.

	C1			C2		
	GA	GA-2	ACO	GA	GA-2	ACO
Lower error (%)	0	3.07	0.61	0	0	1.54
Average (%)	12.77	5.49	17.01	15.45	10.80	10.75
Higher error (%)	49.85	28.90	31.41	55.19	31.74	22.59
Errors above 20%	25.4	0.01	25.8	29.7	9.2	0.6
Average runtime	18sec	18sec	1 sec	18sec	18sec	1 sec

Table 1. Results for datasets C1 and C2

Looking at Figure 1 and first columns of Table 1 we can observe the results for dataset C1. Errors for GA-2 are mainly concentrated in the band of 4% (more than 60% of occurrences). Although for GA the concentration peaks were similar as in GA-2, the concentration was much lower, and the results were much more spread over the observed error bands. For ACO the results were distributed between 10% to 21% ranges.

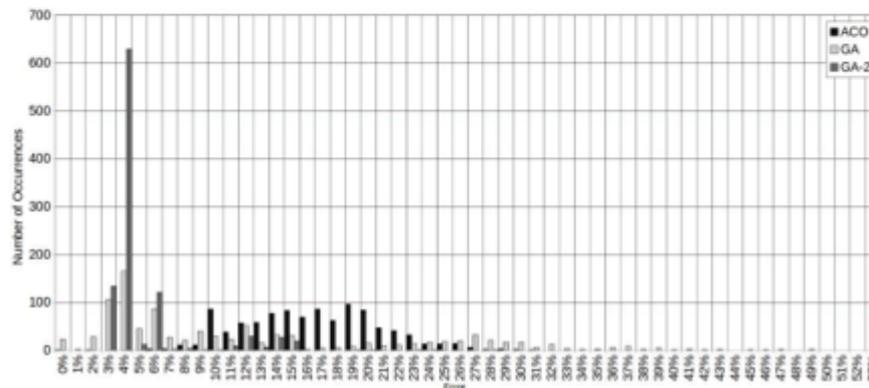


Figure 1. The combined histograms of the results for the datasets C1 - Step 1.

In the final columns of Table 1 are presented results for data set C2¹. For the three methods, the higher concentration of results is between the error bands of 2% and 20%. For ACO, the main concentration points are in 10 and 6% of error bands. For GA-2, the highest concentration of results is in the bands of 2, 5, 6 and 10%. These results are distributed over a much longer interval than in ACO, ranging from 0% to 31%, implying in a lower quality of results than ACO. GA obtained the worst overall quality of results, which are spread over a much longer interval than for other methods (up to range of 55%).

For both datasets, although execution times for GA and GA-2 were considerably longer (about 18 seconds) than for ACO (one second), the times for the three methods were

considered very good. These times allow, for example, multiple executions of the methods during the briefing of the mission, or even run the application during the flight of the aircraft, if it were necessary to draw up a new route. These results point to the feasibility of the methods proposed and implemented in this step.

4. PROPOSED METHODS AND RESULTS - STEP 2

This step includes the restriction for the autonomy limit of the aircraft. This new restriction implies that not all designated targets can be identified without the aircraft having to return to the mother-ship for refueling. Thus, the inclusion of a new goal it is necessary: maximizing the number of targets identified on the basis of autonomy limit, because only minimizing the route does not produce acceptable solutions. Since there is no need to identify all targets, the smallest route would be the one with only the target with shortest total distance, relative to the mother-ship. Following we present the adaptations made in the functions used in Step 1. The complete set of functions adapted and implemented are available at [Yokoyama 2016]. These adaptations include modifications of functions previously used and the inclusion of new functions to meet the new restrictions.

New functions Proposed for the Ant Colony Optimization

The Pheromone Deposit Function was adapted and the amount of pheromone is inversely proportional to the length of each edge (i; j) given by Equation 4:

X is a constant and d_{ij} is the distance of the edge (i; j) traversed by the ant k.

$$\Delta\tau_{ij}^k = \begin{cases} \frac{C}{d_{ij}} & \text{if the edge}(i, j) \text{ is part of the ant } k\text{'s route} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

We proposed two versions of the Exploration Function. For both the route construction is done in a similar way to that seen in step 1 and using the same Equation 1. In the first version, after selecting each target (except for the mother-ship), the total distance of the route is evaluated. If this distance is greater than the maximum autonomy limit of the aircraft, the newly selected target is discarded, with only the previously selected targets remaining, and the route construction process is completed. If the distance is less than the autonomy limit, the route construction process continues until the limit distance is exceeded, or all targets are selected. The total distance of the route is considered the distance to the last selected target plus the return distance to the mother ship. During the construction of the route, the pheromone

deposit is also carried out at the edge between the previously selected target and the newly selected target by means of the Pheromone Deposition Function. At the end of route construction, if it has at least one valid target (other than 0), it is stored until the total number of routes defined for each iteration. This number is equal to the number of targets in the dataset. Then the evaluation function is applied to the generated route set. After the evaluation of the routes the Pheromone Evaporation Function is applied on all the edges. This process is repeated at each iteration. The tests with this version of the Exploration Function are called ACO.

In the second version of the Exploration Function, the check is not performed if the route is still valid after insertion of each new target, as in the first version. As in step 1, new targets are added to the route until all targets in the data set have been added. After the route has been built, with all the available targets, the evaluation process begins. This is an iterative process where the first iteration consists of evaluating the total distance of the route with all steps. For each new iteration, the step number is decreased by 1, ignoring the steps at the end of the route. This process continues until the only remaining target is "0". At each iteration, if the total distance of the route to the last evaluated target exceeds the autonomy limit of the aircraft, this route target is replaced by "0". However, if the total distance is less than or equal to the autonomy, the pheromone deposition function is applied over the edges to the last target evaluated in the iteration. In this way, the invalid portion of the route is deleted and the valid part is enforced. At the end of this process the route is stored and the process of creating a new route is started. This process is repeated until the number of routes stored is equal to the total number of targets as in the first version quoted above. Pheromone Evaluation and Evaporation Functions are then applied. This process is repeated at each iteration until it reaches the maximum number of iterations. Tests with this version of the Exploration Function are called ACO-2.

For the Evaluation Function, unlike step 1, the evaluation is made based on route distance and the number of enlightened targets in the route. So, that the greater the number of targets the better route, given by Equation 5.

$$f_i = ((l_{s_{min}} + l_{s_{max}}) - l_{s_i}) * \left(\frac{s_i}{S}\right) \quad (5)$$

Where l_{s_i} is the average step-distance of the i -th route defined by $l_{s_i} = l_i / s_i$; $l_{s_{min}}$ is the lowest value of l_{s_i} and $l_{s_{max}}$ is the largest value of l_{s_i} in the evaluated route set; l_i is the total

distance of the i -th route; s_i is the number of targets of the i -th route; S is the total number of targets in the input data set.

New functions for the Optimization Based on Genetic Algorithm:

Due to the fact that the parents individuals in this step can have different amounts of targets, the Crossover Function need some adaptations. For Fitness Function, the fitness (defined by Equation 5) is given by the distance of the route represented by each individual and the number of enlightened targets, so that the shorter the route and the greater the number of targets, the better the fitness.

The Mutation function implemented is similar to the one used in step 1 with two alleles are selected to be swapped, but with the first one being selected from the alleles of the individual on which the function is being applied, and then a set of candidate alleles is created. This candidate set includes all targets of the input data set except for target 0 and target represented by the first allele selected for mutation. The second allele is selected from the set of candidates, if this allele is part of the sequence of the individual, the position of the two alleles are swapped. Otherwise, the first allele is replaced by the second allele. In this way it is ensured that no element in the sequence is repeated and also allows elements that were not part of the initial sequence to be included. This function also randomly decides if the mutation operation will occur or not, it may even occur multiple times.

The functions for Individuals Generation, Selection and Evolutions are presented at [Yokoyama 2016].

4.1. RESULTS FOR DATASETS C1 AND C2

The analysis was done based on the percentile of error related to the best solution found for each dataset (C1 and C2). The error is calculated on the normalized value of the fitness, where the highest fitness was considered as 100 % and is given by $e_i = 100 a_i$ where a_i is the normalized fitness value of the evaluated solution.

In Table 2 are related the results for datasets C1 and C2. In this table we can observe that all methods using C1 obtained solutions with 0% error. The ACO-2 method obtained the lowest average error (0.06%). The errors obtained by ACO and ACO-2 was below 6%, while for the GA and GA-2 were above 45%. In addition, only the ACO and ACO-2 methods showed all results below the tolerance of 20% error.

	C1				C2				
	GA	GA -2	AC O	AC O2	GA	GA -2	AC O	AC O2	
Lower error (%)	0	0	0	0	0	0	0	0	
Average (%)	12.1	9	4.87	0.24	0.06	16.8	3.1	0.02	0.032
Higher error (%)	50.8	2	46.9	7	5.11	52.1	38.	5	39
Errors above 20%	24.5	0.3	0	0	0	35.5	0.7	0	0
Average runtime	5sec	5se.	1	1	5sec	5se	1	1	sec.

Table 2. Results for datasets C1 and C2 - Step 2

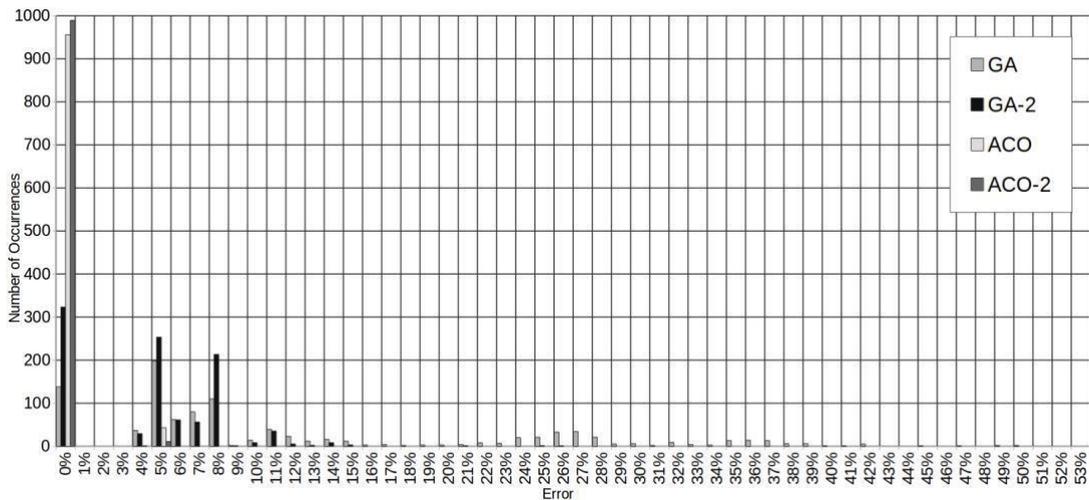


Figure 2. The combined histograms of the results for datasets C1

In the graph of Figure 2 are presented the results for dataset C1. It can be seen that ACO-2 had the concentration point in the range of 0% of error (98.9% of the results), presenting the best overall quality of the results. It is followed by ACO, with 95.6% and GA-2 with 32.3% and GA with 13.8% of the results in the range of 0% of error.

The results for dataset C2 are similar to C1. ACO and ACO-2 have a superiority of the overall quality of the solutions, with respectively, 98.4% and 98.2% of the results in the range 0% error and for both, the error did not exceed 6%. Although the GA-2 obtained a good amount of results in the 0% error range (66% of the results), the results were spread over a larger range of errors, reaching 23% of error. Thus, presenting a general quality lower than ACO and ACO-2, but still being considered good. The GA obtained a small amount of results in the 0% error range, with a considerable peak in the 15% error range, resulting in a very low overall quality.

For both, C1 and C2 (Table 2), the execution times was on average 1 second for ACO and ACO-2 and approximately 5 seconds for GA and GA-2. In general, these times were considered very good for the four methods, allowing for example, multiple execution of the methods during the mission briefing, or even to execute the application in-flight if a new route is required.

5. CONCLUSION

Analyzing the results obtained with the two meta-heuristics, we could conclude that the methods proposed and implemented are good approaches to solve the problem of air reconnaissance of targets. Although both heuristics presented satisfactory results, the use of the methods based on ant colony optimization are the most suitable for the problem addressed in this work. Not only the ACO presented the best overall results but also the best time to solution. However, it is important to note that, although this meta-heuristic has been used quite successfully for optimizing solutions to combinatorial problems, they tend to suffer with “traps” of local optimum depending on the size of the problems. In addition, the time of execution tends to increase due to the size of the problem. The experiments performed in this work were based on 20 targets, but preliminary tests support the analysis presented here for up to 60 targets.

As future works are considered to adapt the methods developed in this work to include three new restrictions (importance, obligatory identification and prioritization of targets) and the optimization of the algorithm through the use of distributed computing.

REFERENCES

Chatterjee, S., Carrera, C., and Lynch, L. (1996). Genetic algorithms and traveling salesman problems. *European Journal of Operational Research*, 93(3):490–510.

Hassan, M. R., Hasan, M. K., and Hashem, M. M. A. (2013). An improved acs algorithm for the solutions of larger tsp problems. *CoRR*, abs/1304.3763.

Li, B., Wang, L., and Song, W. (2008). Ant colony optimization for the ts problem based on ants with memory. In *4th International Conference on Natural Computation*. IEEE.

Wiesebron, M. (2013). Amazonia azul: Pensando a defesa do território marítimo brasileiro. *Rev. Brasileira de Estratégia e Relações Internacionais*, 2(3):107 – 131.

Yang, J., Shi, X., Marchese, M., and Liang, Y. (2008). An ant colony optimization method for generalized tsp problem. *Progress in Natural Science*, 18(11):1417 – 1422.

Yokoyama, A. M. (2016). Avaliação do uso de metaheurísticas baseadas no comportamento da natureza em apoio a operações de esclarecimento por aeronaves de asa móvel. Master's thesis, Laboratório Nacional de Computação Científica. <http://comcidis.lncc.br/downloads/dissertacao/mestrado/tIw2DDb3WqS.pdf>