29th Congress of the International Council of the Aeronautical Sciences
St. Petersburg, Russia; September 7-12, 2014

A MULTI-OBJECTIVE GENETIC ALGORITHM FOR A MAXIMUM COVERAGE FLIGHT TRAJECTORY OPTIMIZATION IN A CONSTRAINED ENVIRONMENT

Bassolillo, S.*, D'Amato, E.*, Notaro, I.*, Blasi, L.*
*** Department of Industrial and Information Engineering (DIII) Second University of Naples (SUN), Via Roma 29, 81031, Aversa, Italy**

Abstract

The problem of generating an optimal flight trajectory with the objective of maximizing the coverage of specified target areas minimizing the total flight path, in the presence of a constrained environment, is solved via multi-objective Genetic Algorithms. A novel coverage model, based on the evaluation of a so called coverage potential field, is proposed. Sensitivity studies with increasing problem complexity allow validating procedure effectiveness as well as final solution reliability.

1 Introduction

The use of unmanned aerial vehicles requires a careful resources optimization to maximize effectiveness and to reduce risks and operational costs. To this end the planning of an optimal flight trajectory, consistent with mission objectives, operational scenario, and vehicle dynamics and performance, plays an important role.

Parameters defining flight missions are usually related to regions to fly over, desired flight altitudes on targets. The operational scenario also provides constraints depending on take-off and landing areas, no-fly zones, the presence of mountains or adverse climatic conditions, minimum/maximum distance from base stations or cooperating vehicles. Finally, constraints related to the specific aircraft used, like maximum climbing rate, maximum and minimum speed, minimum turning radius, range and endurance etc., have to be enforced.

Due to the complexity of the problem, non-conventional, nature-inspired optimization methods received much attention in the last decade. These non-deterministic methods have shown their effectiveness and robustness in a wide range of optimization problems, taking advantage from some specific features such as the capability to handle mixed-type design variables accounting for a large number of constraint functions, and a parallel-like searching method leading to a greater effectiveness in finding global minimum within the design space. An example of hybrid techniques applied to the optimization of a space plane re-entry trajectory problem can be found in [1]. Hybrid soft computing and evolutionary techniques was used in [2] to compute optimum flight path for unmanned air vehicles under several aerodynamic constraints. Different applications of Particle Swarm Optimization algorithms can be found in [3,4,5]. A real-time free flight path optimization based on improved genetic algorithms is reported in [6]. A numerical potential field method combined with a genetic optimizer has been applied for mobile robot path planning in [7].

The objective of this paper is the development of a Genetic Algorithm procedure for flight path optimization compliant with operational constraints aimed at the coverage area maximization and path length minimization. Assuming a typical surveillance mission, environmental constraints are defined in terms of no-fly zones, take-off and landing areas. Flight paths are described through a discrete number of waypoints interpolated by cubic Catmull-Rom splines. Each individual of the genetic population represents a sequence of vectors defining the waypoints. Trajectory starts from a specified point with a given direction and ends on a selected landing area.

Both single-objective and multi-objective optimization procedures have been implemented. The former minimizing total flight path length was aimed mainly at tuning some of the optimization parameters; the latter besides the path length minimization also try to maximize the trajectory length covered over specified target areas. To this end a novel coverage model, based on the evaluation of a so called coverage potential field, is proposed. A noticeable advantage deriving from the application of this model is the possibility to handle target areas of any shape.

Sensitivity studies with increasing problem complexity are performed changing number and position of both targets and no-fly zones. The proposed tests allow validating the procedure effectiveness as well as the final solution reliability.

The paper is organized as follows. In Section 2 basic concepts of genetic algorithms are introduced. In Section 3 the trajectory optimization problem is formulated, then the spline-based approach used in flight path definition and the proposed novel coverage model are described as well.

In Section 4 the proposed algorithm is applied to solve different optimization problems in 2D and 3D scenarios.

Final conclusions and remarks are reported in Section 5.

2 Genetic Algorithm

2.1 Basic Concept

Genetic algorithm (GA) is an optimization technique based on the Darwinian principles of evolution. Inheriting genetics and natural selection paradigms, a GA can describe a set of possible solutions, or *individuals*, making them to evolve towards the optimum maximizing a *fitness* function (i.e. minimizing or maximizing specific objective functions). A finite set of solutions is called *population*.

Such evolution is achieved over following epochs, or generations, until a termination criterion is not met.

Let $X_1, X_2, \ldots, X_n$ be subsets of *n* Euclidean spaces, denoted as search space, with $X_i \subseteq \mathcal{R}, i = 1 \ldots n$.

In this frame, each individual of the population, $\underline{s}$, is characterized by a *n*-tuple of design variables, $\underline{s} = (x_1, x_2, \ldots, x_n)^T$, where $x_i \in X_i,\ i = 1, \ldots n$.

As in nature each individual differs from another one thanks to its own DNA, in the algorithmic counterpart each individual is represented by a string (chromosome) of *n* binary-coded numbers (genes) each related to a specific design variable (Fig.1).

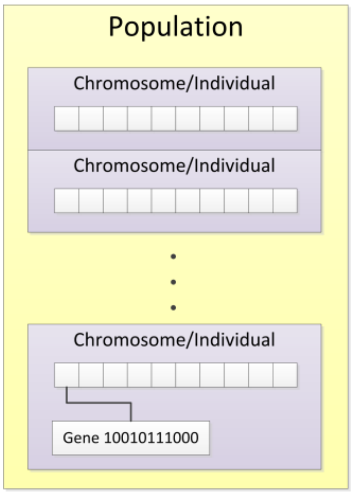


Fig. 1. Population encoding in a GA

The genetic algorithm consists of several steps, as shown in Fig. 2.

- Population initialization: a set of solutions are randomly generated in the search space.
- Fitness computation: at this step objective function and constraints are evaluated to sort individuals and get the best solution (Fig. 3).
- Termination criterion: usually two criteria are defined in a GA, the first one based on the maximum number of generations and the second one based on the maximum number of generations without improvements of the best fitness. In this paper the first criterion has been used.

- Selection: couples of parents are selected for mating among the sorted population on a probabilistic basis. Two selection methods have been implemented: roulette wheel and tournament based. According to the roulette wheel selection method, each individual has a probability to be chosen proportional to its fitness. As for the tournament-based selection, the same criterion is applied on population subsets in a sort of knockout tournament.

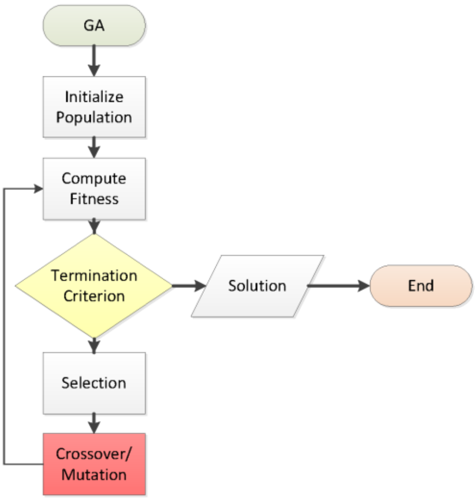


Fig. 2. Genetic algorithm flowchart

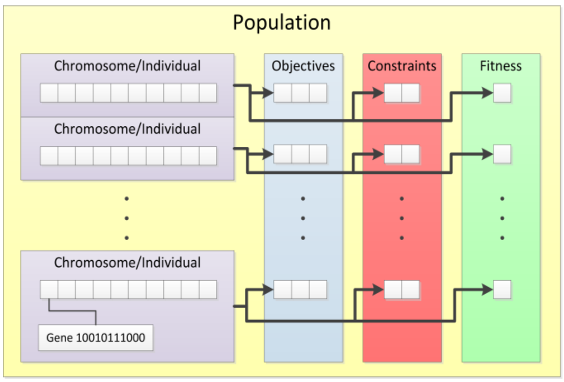


Fig. 3. Fitness computation

- Crossover: on selected parents, a binary crossover operator is applied to create two new individuals (Fig. 4).
- Mutation: to avoid premature stagnation of the algorithm a mutation operator is used, randomly changing a bit of the just created chromosomes. Mutation probability can be a variable parameter during evolution.

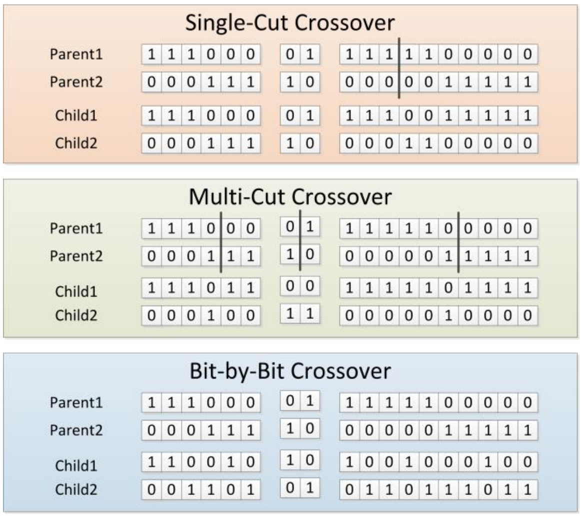


Fig. 4. Crossover operator

3 The optimization problem formulation

3.1 Problem description

The optimization problem consists in finding the shortest path starting from a take-off point and ending on a selected destination point compliant with operational constraints due to environment (e.g. obstacles) as well as specific aircraft performance (e.g. maximum turn radius).

Moreover, to completely plan a typical UAV surveillance mission, one or more coverage areas to explore or supervise must be defined too [8]. That's why both the total trajectory length and the covered area have been included in the selected figure of merit (fitness) of the optimum solution.

To this end, the following objective functions have been defined: the first one (f_1) relates to the trajectory length to be minimized, the second one (f_2) relates to the coverage of target areas to be maximized. In particular, a coverage area index has been defined ranging from 0 (maximum coverage) to 1 (no coverage).

$$f_1 = length(S) \tag{1}$$

$$f_2 = \frac{area_{tot} - area}{area_{tot}} \tag{2}$$

where $area_{tot}$ is the total area to be covered whereas $area$ is the area supervised by the aircraft, taking into account the on-board camera cone of vision.

The fitness function (*fit*) to be maximized has been defined as follows:

$$fit = \frac{c_1}{f_1} + \frac{c_2}{f_2 + 1} \tag{3}$$

where c_1 and c_2 are user-defined weights balancing objective functions f_1 and f_2 of the optimum solution.

As usual, the constrained optimization problem has been treated as an unconstrained one by means of proper defined penalty functions that degrade fitness value whenever one or more constraints are violated. The penalized fitness $\overline{fit}_i$ related to the *i*-th trajectory (or individual) can be calculated as:

$$\overline{fit}_i = \frac{fit_i}{\prod_{r=1}^{P} K_r} \tag{4}$$

where P is the total number of constraint functions. In particular, two constraint functions (P=2) have been considered: one related to the maximum flight path curvature and one related to the presence of obstacles or no-fly zones.

Let $c(S) = \max \left| \nabla \cdot \left(\frac{\nabla S}{\|\nabla S\|}\right) \right|$ be the maximum curvature of the generic path and c_{max} the curvature maximum allowable value. According to eq. (4), coefficient K_1 related to the first constraint can be defined as:

$$K_1 = \begin{cases} p\left(\frac{c(S)}{c_{max}}\right)^{\gamma} & c(S) > c_{max} \\ 1 & c(S) \le c_{max} \end{cases} \tag{5}$$

As for obstacle avoidance, let $g(S)$ be the number of obstacles the generic path passes through. Coefficient K_2 related to the second constraint can be defined as:

$$K_2 = \begin{cases} p(g(S)+1)^{\gamma} & interference \\ 1 & no\ interference \end{cases} \tag{6}$$

where $p \in [1, +\infty]$ and $\gamma \in [1, +\infty]$ are user-defined coefficients enforcing the strengthen of the penalty function.

It is worth noticing that possible safety margins can be easily taken into account properly changing the obstacles border.

3.3 Waypoints position model

Let $\underline{A}=(x_A, y_A)^T$ and $\underline{B}=(x_B, y_B)^T$ be the starting and ending point respectively. Let $X \times Y = [x_A, x_B] \times [y_A, y_B]$ be the search space and $\underline{p_i} = (x_i, y_i)^T \in (X \times Y)$ the *i*-th waypoint with $i = 1, \ldots n$ such that $\underline{p_0} = \underline{A}$ and $\underline{p_{n+1}} = \underline{B}$

For trajectory planning, waypoints position cannot be generated using an *absolute* position model related to a *global* reference frame (o_G,x_G,y_G), because a lot of unfeasible trajectories would result, spending a lot of time to compute useless solutions. To improve the algorithm effectiveness, a *relative* position model has been used by introducing a more suitable *local* reference frame (o_L,x_L,y_L).

Let $M \times \Phi = [0,1] \times [0, \pi]$ and $(m_i, \phi_i) \in (M \times \Phi)$ be the search space and design variables respectively. The *i*-th waypoint position, $\underline{p_i} = (x_i, y_i)^T$, can be computed starting from waypoint $\underline{p_{i-1}}$ as:

$$\underline{p}_{\mathrm{i}} = \underline{p}_{\mathrm{i}-1} + m_{\mathrm{i}} \begin{Bmatrix} \cos\alpha_i \\ \sin\alpha_i \end{Bmatrix} \left(\underline{p}_{\mathrm{end}} - \underline{p}_0\right) \tag{7}$$

with $\alpha_i = \alpha_{min} + \phi_i(\alpha_{max} - \alpha_{min})$

where α_{min} and α_{max} are computed once the local reference frame is set by tracing the line joining the waypoint $\underline{p_{i-1}}$ (center of the local reference frame) and the destination point $\underline{p_{end}}$ (Fig. 5).

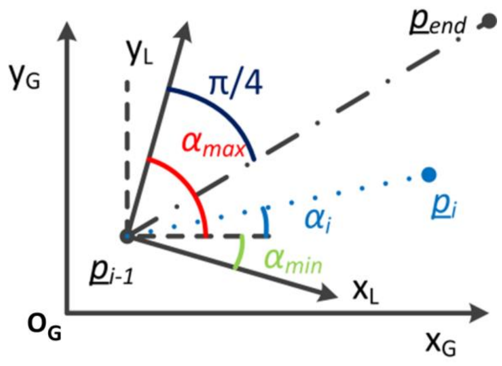


Fig. 5. Waypoint position model

With such a type of waypoints selection algorithm will be forced to trace trajectories going towards the destination point, thus improving its effectiveness.

3.2 Potential field-based coverage model

The basic idea of this new model consists in taking into account the presence of one or more target areas by means of a potential field built at the beginning of the optimization task. This potential field generates an attractive action in the neighborhood of each target area that can be used to modify spline control points selection.

To simplify definition of the coverage area as well as computation of the related potential field, only circular areas are considered. Nevertheless, this model can be applied to areas of any shape, as will be shown in paragraph 4.4. Potential field $F(x,y)$ is defined as:

$$F(x,y) = \frac{1}{\left(1 + \left(r * d(x,y)\right)^2\right)^2} \tag{8}$$

where $d(x,y) = \sqrt{(x-x_c)^2 + (y-y_c)^2}$ is the distance of the arbitrary point (x,y) from the center (x_c, y_c) of the circular target area with radius r.

To increase the probability of paths passing over the target area, equation (7) is modified as follows:

$$\underline{p}_{\mathrm{i}} = \underline{p}_{\mathrm{i}-1} + m_{\mathrm{i}} \begin{Bmatrix} \cos\alpha_i \\ \sin\alpha_i \end{Bmatrix} (\underline{p}_{\mathrm{end}} - \underline{p}_0) \tag{9}$$

with $\alpha_i = \alpha_{min} + \phi_i(\alpha_{max} - \alpha_{min})^{\frac{\nabla_x F}{\nabla_y F}}$

where $\nabla_x F$ and $\nabla_y F$ denote gradients of function $F(x,y)$ computed respectively on the local reference frame axes x_L and y_L, centered in p_{i-1} (Fig. 5).

3.3 Spline-based trajectory definition

Once waypoints are chosen on the search domain, aircraft trajectory is built by means of a Catmull-Rom spline [9]. This family of cubic interpolating splines ensuring C^1 continuity is formulated such that the tangent at each point $\underline{p}_i$ is calculated using the previous and the next control points, i.e. $\tau(p_{i+1} - p_{i-1})$. The geometry matrix is given by:

$$\underline{S}_i(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau-3 & 3-2\tau & -\tau \\ -\tau & 2-\tau & \tau-2 & \tau \end{pmatrix} \begin{pmatrix} \underline{p}_{i-2} \\ \underline{p}_{i-1} \\ \underline{p}_i \\ \underline{p}_{i+1} \end{pmatrix} \tag{10}$$

The τ parameter, known as "tension", allows to control the sharpness of the curve bending at the interpolated control points. In this paper a value of 0.5 is set.

Depending on the specific operational scenario, tangents at initial point, $\underline{p}_0$, and destination point, $\underline{p}_{end}$, are defined by the user.

Thus equation (10) allows to trace the aircraft trajectory interpolating n waypoints $\underline{\tilde{p}}_1, \underline{\tilde{p}}_2, \ldots, \underline{\tilde{p}}_n$ whose position is frozen at the end of the optimization task.

4 Applications and results

Both single-objective and multi-objective optimization procedures have been implemented and applied to different scenarios. The former aimed at the minimization of the flight path length has been preliminarily carried out to tune the optimization parameters. The latter tries also to maximize the coverage of specified target areas. Only normalized dimensions have been used throughout the examples presented in this chapter.

4.1 Single-objective application - Scenario 1

Consider a rectangular area $(x,y) \in [0,1] \times [0,1]$ with the starting point A placed at $(0,0)$ and the destination point B placed at $(1,1)$. 19 circular obstacles of radius 0.05 are placed in a grid pattern resembling in same way a sort of urban scenario (Fig. 6).

Path length has been selected as the objective function to be minimized (f_1). As previously said this application was mainly aimed at the optimization parameters tuning. Selected values are summarized in Table 1.

Table 1. GA parameters (Single-objective application)

Parameter	Value
Population size	100
Generations	100
Crossover probability (%)	100
Crossover type	Multi-cut
Mutation probability[1]	0.01-0.1
Max curvature	50
Penalty function parameter, p	4
Penalty function parameter, γ	3
Fitness weight, c_1	1
Fitness weight, c_2	0

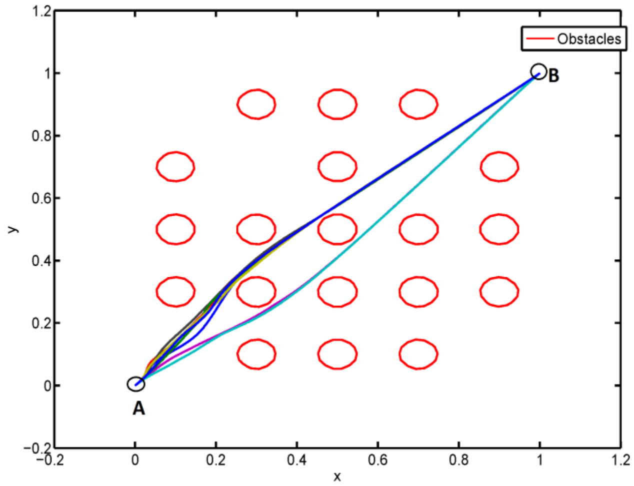


Fig. 6. Scenario 1 – Optimized paths over 8 runs (min length)

In Fig. 6 best paths computed over 8 runs are shown. As can be seen, optimized paths appears quite similar in terms of both shape and length. Table 2 summarizes results in terms of minimum length (best value) and average length, showing a good reliability as well.

Table 2. Single-objective optimization results

	Value
Min path length	1.4226
Average length over 8 runs	1.4245±0.0021
Computational time (s)[2]	29.5

Fig.7 shows the population-averaged objective function f_1 evolution obtained over 8 runs. The selected number of epochs is quite enough to provide a good convergence of the path length to the minimum; as can be seen, the algorithm actually takes less than 20 epochs to identify the optimum solution.

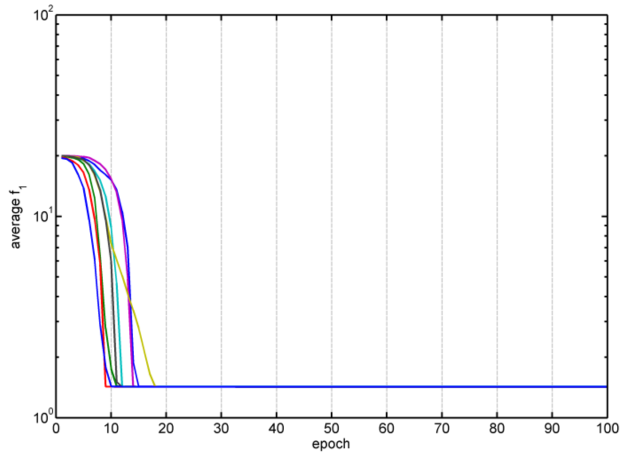


Fig. 7. Scenario1 - Average objective function f_1 evolution over 8 runs

4.2 Multi-objective application - Scenario 2

This scenario is obtained adding 1 circular target area centered at point $(0.3, 0.7)$ with radius 0.1 to Scenario 1. To compute the related potential field, searching domain has been meshed using a (50×50) grid with 2500 elements.

The minimum path length maximizing the coverage of target area has to be identified by the algorithm. As for optimization parameters, same values used in the previous application have been selected (see Tab. 1) with the exception of fitness weights ($c_1 = c_2 = 1$).

Fig. 8 shows best paths obtained over 8 runs. Also in this application a sufficient reliability was obtained showing most of the paths a similar shape. Table 3 summarizes results related to the best solution as well as the average one over 8 runs.

Evolution of both objective functions, f_1 and f_2 , related to the best run are shown in Fig. 9. As can be seen, this applications is much more demanding in terms of number of generations: the algorithm takes almost all the 100 epochs to find the optimal path.

[1] Mutation probability is linearly increased during the evolution to avoid premature stagnation of the algorithm
[2] Test on an Intel Core i3 @1.3GHz, 8GB RAM

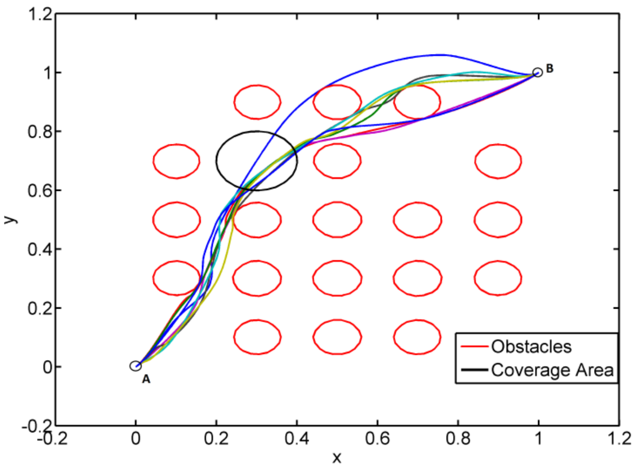


Fig. 8. Scenario 2 – Optimized paths over 8 runs (min length + max coverage)

Table 3. Multi objective optimization results (Scenario 2)

		Value
Best solution:	length	1.5180
	coverage	0.3707
Average solution:	length	1.5525±0.0432
(over 8 runs)	coverage	0.3793±0.0276
Computational time (s)		46.4

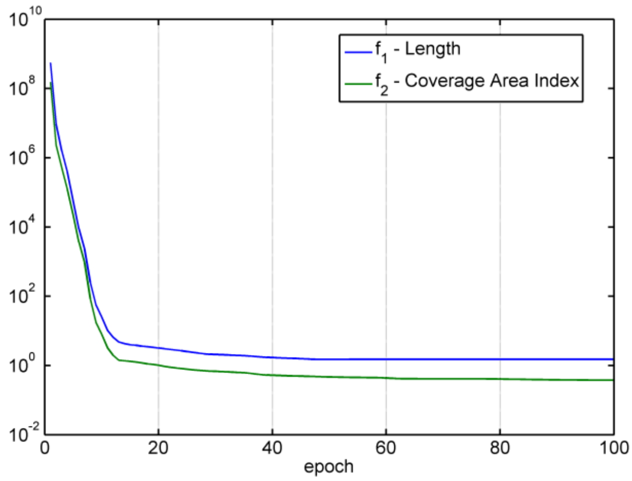


Fig. 9. Scenario 2 - Objective functions f_1 and f_2 evolution

4.3 Multi-objective application - Scenario 3

This scenario was obtained adding 1 more circular target area centered at point $(0.7, 0.7)$ with radius 0.1 to Scenario 2. In order to test the optimization procedure sensitivity to different fitness weights values, three optimization tasks have been performed balancing in a different way objective functions f_1 and f_2. The first one, $(c_1 = 0\,; c_2 = 1)$ was aimed at finding the trajectory with the minimum coverage index (i.e. maximum coverage area), the second one, $(c_1 = 1\,; c_2 = 1)$ was aimed at finding the mean solution equally balancing path length and coverage area. Finally the third one, $(c_1 = 1\,; c_2 = 0)$ was aimed at finding the minimum length trajectory.

Table 4 summarizes GA parameters and final results obtained at the end of the three different optimization tasks in terms of path length and coverage index.

Table 4. Multi-objective optimization – GA parameters and results (Scenario 3)

Parameter		Value
Population size		100
Generations		200
Crossover probability (%)		100
Crossover type		Multi-cut
Mutation probability		0.01-0.05
Nr. Waypoints		10
Max curvature		50
Penalty function parameter, p		4
Penalty function parameter, γ		3
Max coverage path:	length	1.6233
	coverage	0.0953
Mean solution:	length	1.4745
	coverage	0.5000
Min length path:	length	1.4281
	coverage	0.6286

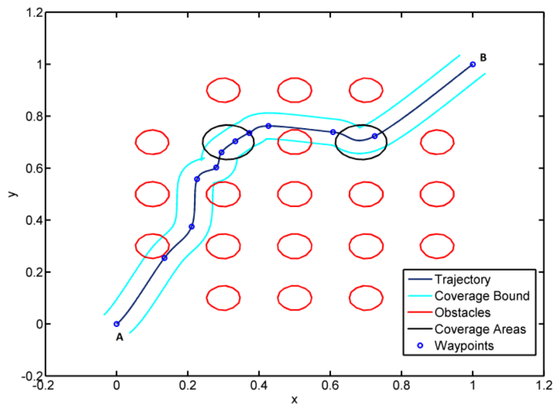
Optimized trajectories related to the three different tasks are shown in Fig.11.

It is worth noticing that the minimum length path is consistent with the single objective solution previously shown in Section 4.1. Nevertheless, compared with Scenario 2, a double number of generations was generally required to solve a multi-objective optimization problem with this more complex scenario.

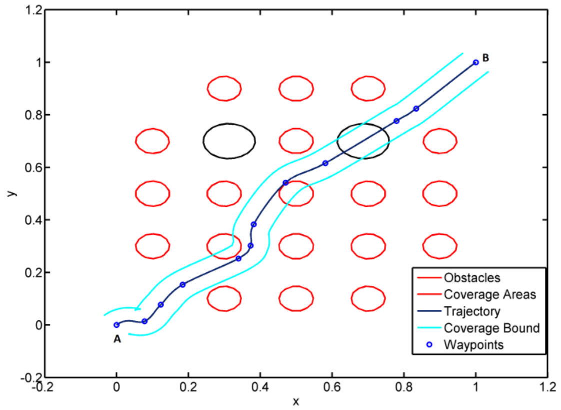
4.4 Multi-objective application - Scenario 4

A specific scenario with non-convex obstacles has been defined to assess algorithm capability in finding optimum path through obstacles with a more general shape.

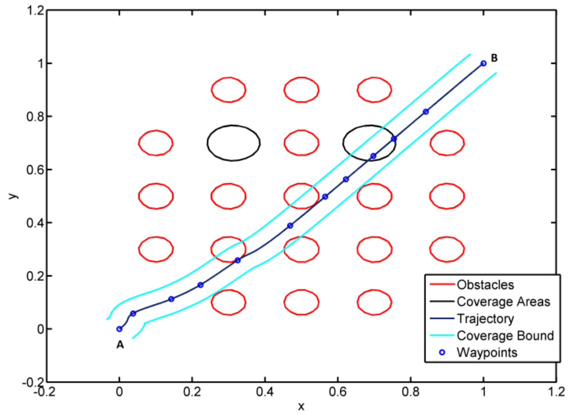
A rectangular area $(x, y) \in [0,1] \times [0,1]$ with the starting point placed at (0,0) and the destination point placed at (0.8,0.8) has been considered.



Maximum coverage solution $(c_1 = 1 \,; c_2 = 0)$



Mean solution $(c_1 = 1 \,; c_2 = 1)$



Minimum length solution $(c_1 = 0 \,; c_2 = 1)$

Fig. 11. Scenario 3 – Optimized paths

Two non-convex obstacles have been placed close to the starting point and destination point respectively. Moreover, a target area has been placed inside the concavity of the obstacle near the destination point to further increase the complexity of this scenario (Fig. 12).

Objective functions f_1 and f_2 have been equally balanced setting $c_1 = 1$ and $c_2 = 1$.

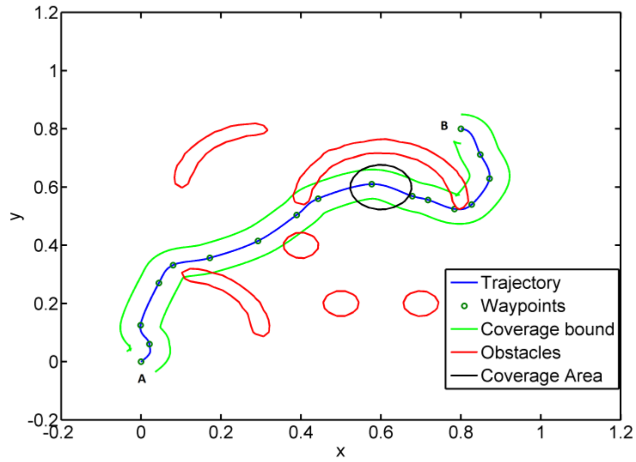


Fig. 12. Scenario 4 – Optimized path

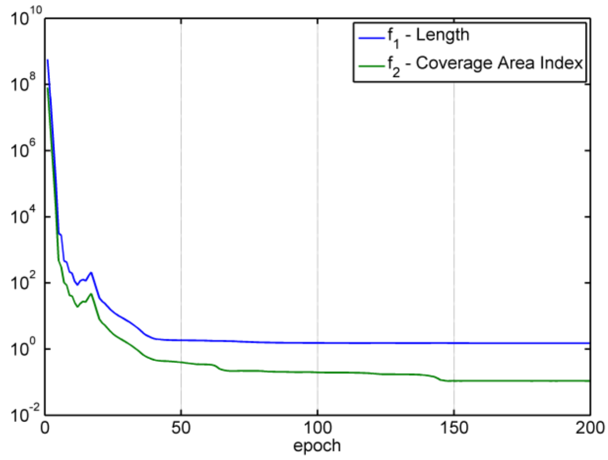


Fig. 13. Scenario 4 - Objective functions f_1 and f_2 evolution

Table 5. Multi-objective optimization – GA parameters and results (Scenario 4)

		Value
Population size		200
Generations		200
Crossover probability (%)		100
Crossover type		Multi-cut
Mutation probability[3]		0.01-0.05
Nr. Waypoints		15
Max curvature		50
Penalty function parameter, p		5
Penalty function parameter, γ		8
Computational time (s)		187
Best solution:	length	1.5160
	coverage	0.0870

[3] Mutation probability is linearly increased during the evolution to avoid premature stagnation of the algorithm

Optimum path is shown in Fig. 12. As can be seen the algorithm was able to find a way out the concave obstacle reaching the destination point *B*. Evolution of objective functions f_1 and f_2 is shown in Fig. 13. As can be seen the selected number of generations allows a satisfactory convergence to the optimum solution. In Table 5 selected GA parameters and results are summarized.

4.5 Multi-objective application - Scenario 5

To assess algorithm capability to solve three dimensional real-world problems an extension of waypoints relative position model described in Section 3.3 is proposed.

Let $M \times \Phi \times \Theta = [0,1] \times [0,\pi] \times [0,\pi]$ and $(m_i, \phi_i, \theta_i) \in (M \times \Phi \times \Theta)$ be the search space and design variables respectively. Position of the *i*-th waypoint, $\underline{p_i} = (x_i, y_i, z_i)^T$, can be computed as:

$$\underline{p}_i = \underline{p}_{i-1} + m_i \begin{Bmatrix} \cos\alpha_i \\ \sin\alpha_i \\ \sin\beta_i \end{Bmatrix} (\underline{p}_{end} - \underline{p}_0) \qquad (11)$$

whit $\alpha_i = \alpha_{min} + \phi_i(\alpha_{max} - \alpha_{min})^{\frac{\nabla_x F}{\nabla_y F}}$

$$\beta_i = \beta_{min} + \theta_i(\beta_{max} - \beta_{min})$$

where β_{min} and β_{max} are computed once the local reference frame is set by tracing the line joining the waypoint $\underline{p_{i-1}}$ (center of the local reference frame) and the destination point $\underline{p_{end}}$ (Fig. 14).

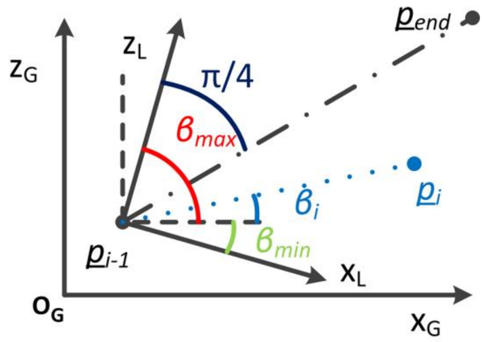


Fig. 14. Extended waypoint position model

Let $(x, y, z) \in [0,1,1] \times [0,1,1]$ be the search space with the starting point *A* placed at (0.2,0.2,0) and the destination point *B* placed at (0.8,0.8,0). 13 obstacles are placed like a sort of mountain scenery with the target area placed in a valley (Fig. 15). Objective functions f_1 and f_2 have been equally balanced setting $c_1 = 1$ and $c_2 = 1$.

Table 6. Multi-objective optimization – GA parameters and results (Scenario 5)

		Value
Population size		1000
Generations		500
Crossover probability (%)		100
Crossover type		Multi-cut
Mutation probability[4]		0.01-0.05
Nr. Waypoints		7
Max curvature		50
Penalty function parameter, *p*		2
Penalty function parameter, *γ*		3
Computational time (s)		2157
Best solution:	length	2.3520
	coverage	0.7013

Table 6 summarizes selected GA parameters and results whereas the optimized path is shown in Fig. 15.

As we can see the algorithm was able to find a feasible trajectory passing over the specified target area and reaching the destination point. As expected, to maximize coverage area balancing at the same time path length, the algorithm tried to selected the altitude as high as possible just over the target being coverage bound the result of the flight altitude.

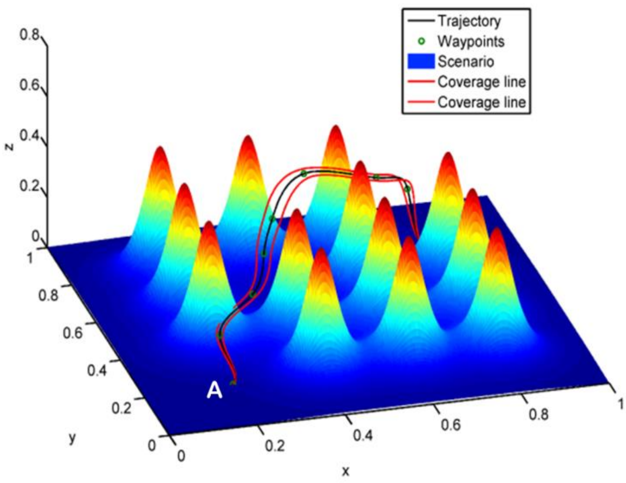
The proposed approach appears to be effective also in a three dimensional scenario. On the other hand a considerable increase in the computational effort resulted compared with previous applications even though a quite simple scenario has been defined.

4.6 Multiple trajectories optimization

Possibility to identify multiple trajectories has been deemed an interesting problem to be investigated especially when a single aircraft could be not sufficient to cover large areas. In particular the optimization procedure has been applied to identify two trajectories minimizing total length and maximizing total coverage of

[4] Mutation probability is linearly increased during the evolution to avoid premature stagnation of the algorithm

specified target areas. Same scenario used in Section 4.3 (Scenario 3) has been considered.



TOP VIEW

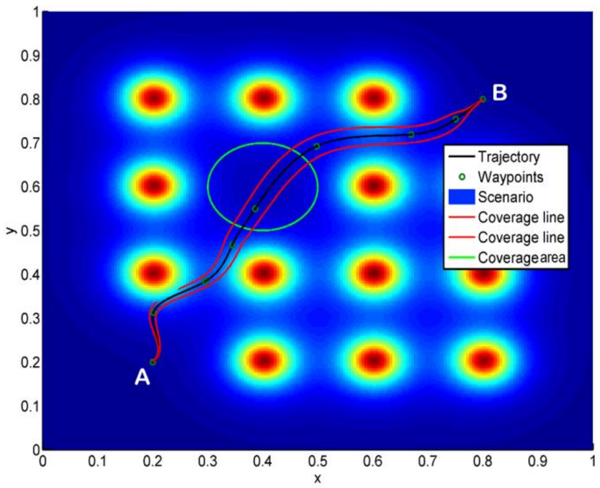


Fig. 15. Scenario 5 – 3D Optimized path

As this preliminary application concerns an off-line path planning, it is worth emphasizing that no aircraft collision avoidance model has been developed for the time being.

Fitness function has been defined equally balancing objective functions f_1 and f_2 ($c_1 = 1\,; c_2 = 1$); moreover to take into account multiple trajectories both functions f_1 and f_2 have been modified as follows:

$$f_1 = length(S_1) + length(S_2) \tag{12}$$

$$f_2 = \frac{area_{tot} - area_1 - area_2}{area_{tot}} \tag{13}$$

where S_1 and S_2 are the two different paths, $area_1$ is the area covered by the first aircraft and $area_2$ is the area covered by the second aircraft.

In order to properly evaluate the objective function f_2, any possible overlap of the second aircraft trajectory just over the target area do not contribute to $area_2$ increment.

In Table 7 selected GA parameters and results are summarized whereas the best path is shown in Fig. 17.

Table 7. Multiple trajectories optimization – GA parameters and results

Parameter		Value
Population size		500
Generations		1000
Crossover probability (%)		100
Crossover type		Multi-cut
Mutation probability		0.01-0.05
Nr. Waypoints		6
Max curvature		50
Penalty function parameter, *p*		4
Penalty function parameter, *γ*		3
Computational time (s)		2543
Best solution:	length	1.5709+1.4408
	coverage	0.1365

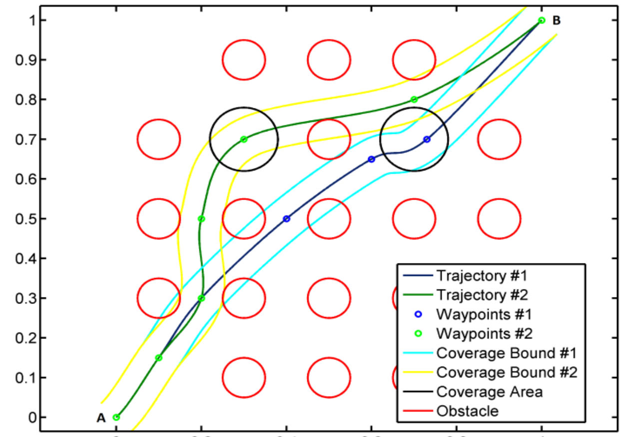


Fig. 17. Multiple optimized trajectories

As can be seen the final solution appears much more effective in terms of coverage index compared with the *mean* solution obtained in Section 4.3 (Tab. 4) and rather comparable with the *max coverage* solution. On the other hand this application appears much more demanding in terms of both population size and generations number.

5 Conclusions

In this paper, a Genetic Algorithm procedure for off-line flight path optimization has been presented.

Flight paths have been described through a discrete number of waypoints interpolated by means of cubic Catmull-Rom splines. Path length and coverage area are the selected objective functions to be minimized and maximized respectively.

Assuming typical surveillance missions, different scenarios have been defined to test the procedure effectiveness in finding near-optimal solutions compliant with operational constraints. Results show the algorithm is able to provide quite good solutions also in the presence of non-convex obstacles.

Finally a preliminary assessment of optimization procedure capability in handling multiple trajectories has been carried out. Such capability could represent an important feature usable whenever mission effectiveness is affected by poor aircraft range performance and it should be further investigated with much more complex scenarios.

As for computational effort, additional work should be devoted to reduce computational time in order to make this procedure suitable for possible 3D on line application. For the time being it could be rather used as a pre-flight path planner working with a real time optimizer devoted to the refinement of such pre-defined trajectories.

References

[1] N. Yokoyama, S. Suzuki, Modified Genetic Algorithm for Constrained Trajectory Optimization, *Journal of Guidance, Control and Dynamics*, Vol. 28, 2005, pp. 139-144.

[2] R. Vaidyanathan, C. Hocaoglu, T.S. Prince, R.D. Quinn, Evolutionary Path Planning for Autonomous Air Vehicle Using Multiresolution Path Representation, *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, Maui, HI, USA, Oct. 29-Nov. 3, 2001.

[3] L. Blasi, S. Barbato, and M. Mattei, A particle swarm approach for flight path optimization in a constrained environment, *Aerospace Science and Technology*, Vol. 26, n.1, 2013, pp.128-137.

[4] Raja, P. and Pugazhenhi, S., Path Planning for Mobile Robots in Dynamic Environments using Particle Swarm Optimization. *Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing*, Kottayam, Kerala, India, 2009, pp. 401-405

[5] Wang L., Liu Y., Deng H. and Xu Y., Obstacle-avoidance Path Planning for Soccer Robots Using Particle Swarm Optimization. *Proceedings of IEEE International Conference on Robotics and Biomimetics*, Kunming, China, 2006, pp. 1233-1237.

[6] X.B. Hu S.F. Wu J. Jiang, On-line free-flight path optimization based on improved genetic algorithms, *Engineering Application of Artificial Intelligence*, Vol. 17, 2004, pp. 897-907.

[7] L. Lei, H. Wang, Q. Wu, Improved Genetic Algorithms Based Path Planning of Mobile Robot Under Unknown Environment, *Proceedings of IEEE International Conference on Mechatronics and Automation*, Luoyang, China, June 25–28, 2006, pp. 1728-1732.

[8] E. D'Amato, Multiobjective evolutionary-based optimization methods for trajectory planning of a quadrotor UAV, *PhD Thesis, 3D Tech*, 2013.

[9] Catmull E., and Rom R., A class of local interpolating splines, In: *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Reisenfeld, Eds. Academic Press, New York, 1974, pp. 317–326.

8 Contact Author Email Address

mailto: egidio.damato@unina2.it

Copyright Statement