Software Engineering Department

ORT Braude College

Capstone Project Phase A

Research on Path Planning Algorithm for Multi-UAV Targets Based on Genetic Algorithm

Project number : 22-1-R-25

Supervisor: Prof. Miri Weiss-Cohen

Ajwad Zidan

zidanajwad@gmail.com

Mahmood Odeh

odeh.mahmood3@gmail.com

Abstract:

In recent years, the use of UAVs for various purposes has increased in many fields like: military uses, research and others. While at the same time the problems and challenges increase parallel with the increase in use. One of the common problems in UAVs use is path planning. In path planning the goal is to calculating paths at minimal costs, finding the shortest path and accuracy. Many researches have proposed different solutions, many of these solutions have based on the Genetic Algorithm. Despite the many studies and solutions, the problem has not yet completely solved. In this study we propose a solution to the problem of path planning based on Genetic Algorithm for multi-UAV maritime targets search, which is similar to the multi-traveling salesman problem (MTSP). Trough the ability to change the operators of the genetic algorithm and makes mutations in order to achieve an optimal solution with less calculations and costs.

1.Introduction:

An unmanned aerial vehicle (UAV) is an aircraft without any human participation physically. UAVs are a component of an unmanned aircraft system (UAS), which include additionally a ground-based controller, a system of communications and a lot of sensors like: cameras, Temperature sensors, thermal Sensors, Ground Penetrating Radar (GPR) and others, to gather information, date back to the early 1900s. UAVs can be different in purpose of use, size, sensors and support systems. UAVs have undergone tremendous development momentum in recent years and play a vital role in military and civilian applications where UAV or groups of UAVs used for hitting enemy targets ,climate monitoring, environmental research, weather forecasting, rescue and search operations. Military UAVS have come to revolutionize warfare [1].UAVs becoming one of the main weapons integrated into military forces throughout the world. The origins of the military UAV date back to the Second World War, with the development of “teletanks” such as the Russian TT-26 and the German Goliath. Both could be driven towards targets and remotely detonated. Military UAVS systems have become indispensable tools for modern armies too. Furthermore, they are used to gather intelligence and information from enemy land, thanks to sensors installed in the UAV like a camera and the fact that the UAV can be operated independently, and the ability to calculate a new path in case an attack or obstacle is detected without risking pilot life. In research and development field: Scientists use UAVS to gather different types of data related to the ground, sea and air thanks to sensor systems installed in the UAV. They can find useful data without sending several teams to the target locations which saves a lot of expenses. UAVS provide accurate scientific data from various locations can be collected quickly and easily. In the field of rescue, especially rescue of lost marine targets due to ship distress accidents, with the increase in the number of maritime transport ships, the number of ship distress accidents has gradually increased. UAVs help rescue and determine the location of marine accidents and survivors. The rise in the use of UAVs development and research adds many and not simple challenges that need to be addressed, in every field of UAV use. Compute a path planning is one of these challenges. The main objective of compute path planning is to have less computational cost and time for computing an optimal path planning. The path generated should be optimal so that it takes less time for computed and minimal path length. This challenge has been extensively researched in the last decade with the increase of using UAVs for various purposes.[2] Furthermore, UAV path planning design a flight path directed to the target with minimal comprehensive costs, i.e., minimal probability of being destroyed while meeting the UAV performance requirements, be accurately, appropriate speed and compliance with environmental conditions and types such as: 2D or 3D environment, an environment with obstacles that known in advance or not, online and offline path planning. UAVs can fly in space by using pre-planned (offline) routes or might be the dynamic route (online). When the environment is dynamic, multiple obstacles exist during the execution of task which added more challenges of compute optimal path planning. Path planning becomes more difficult task for UAVs when the environment is 3D. In 3D environment a lot of challenges and uncertainties comes into picture, one classical problem is taking the UAV physical constraints, like the maximum height can UAV get to, into consideration to control the UAV height limitation and plan non-damage path [3]. Choosing a route planning algorithm that can handle all these challenges could be a complexity task. Despite the progress of development and research to address the challenges of UAVs path planning and provides significant improvement, there is no definitive solution and as the use of UAVs increases, the challenges and problems increases and become more complex. Relying on studies presented in related works in next, we try to offer a solution to the problem of UAV path planning, using scientific and programming tools, for get optimal UAV path.

2.RELATED WORKS

The problem of optimizing UAV Path Planning is solved and analyzed differently among researchers. In [2] Lin Li proposes an improved route planning algorithm based on genetic algorithm to solve the route planning problem of multi-UAV maritime targets search, leaning on emergency equipment that installed with position indicators, which can provide accurate targets positions to be rescued, and make the mission of search overboard maritime targets rapid and accurate. Similar to MTSP problem says Lin [2], searching route planning for overboard maritime targets can be abstracted and resolved by two sub algorithms: Sub-algorithm of marine search and rescue target classification based on K-means and GA, and sub-algorithm for maritime targets path planning based on GA. Lin Li takes into account the change of location of the drift targets under the influence of natural factors like wind and waves and also the characteristics of the target like size and shape. The first sub-algorithm K-means clustering algorithm based on Genetic Algorithm proposed for clustering targets, using the following operators of GA in order to get best and optimal classification: First cluster the set of targets into clusters, where the clusters number is equal to UAVS number, using the positions of the cluster centers to represent the chromosome (Encoding). Construct the fitness function using the criterion function of K-means, which reflects the accuracy of the clusters. Selection operator as combination of save the optimal individuals strategy and randomly select individuals by the roulette. Crossover: an arithmetic crossover method suitable for floating-point number encoding, and linearly combine two paired chromosomes to generate two new offspring chromosomes. Uniform mutation operator to select the mutation point and generate random numbers from the value range. The second sub-algorithm for maritime targets path planning based GA, using multi-chromosome coding where Each gene on the chromosome represents a target, fitness function based on the mean-variance model the mean (average) value of the flying distance of the UAV is taken as the primary objective function, and the variance of the flying distance of the UAV is selected as the secondary objective function. For the selection operator also using a combination of save the optimal individuals and randomly select individuals by the roulette. Sorting Crossover method is adopted for the random selection of gene fragments for the crossover operation. And for mutation Using gene segment insertion. The algorithm proposed by Lin Li in [5] had coding in JAVA language for implement and simulation with 3 UAVs and 30 targets. The results (see Table 1) show that the distance traveled by each UAV is reduced relative to the initial distance, and the difference between the three UAVs is relatively small.

Simulation Results	UAV-1	UAV-2	UAV-3	μ	σ^2
Initial distance	2778	2382	2373	1658	316
Shortest distance	1645	1678	1650		
Number of iterations	120	147	113		

Table 1: The output result of the algorithm proposed by Lin In[5]

In [3] the authors propose a path planning algorithm for UAV based on fusion of improved A* and DWA algorithms. The standard A* algorithm is improved based on UAV'S flight environment information and concave obstacles , through introduce the obstacle weight coefficient into the heuristic function where the obstacle weight coefficient expresses the obstacles map complexity , with determining safety distance from obstacles, which optimizing global paths. Additionally adjustment of the speed evaluation function of DWA , that considers the density of obstacles, safety threshold and can treat well with unknown obstacles . For simulation experiment and analysis, the authors used a 50X50[km^2] planning area with grid processing of arc obstacles and convex processing of concave obstacles that were conducted in MATLAB programing language, with the setting of constraints and restrictions for changing flight direction and safety distance from obstacles. The experiments results compared the efficiency of the algorithms with known obstacles and unknown obstacles to obtain the planning time and the resulting path's length. Compared the improved A* to the standard A* , the path planned by the improved A* is smoother and safety motion guaranteed. Although the simulation results show the efficiency of the improved A * algorithm compared to both DWA and fusion algorithms, shorter planning time and optimal global path length, it is unable to deal with unknown obstacles. Additionally the results show that (see Figure 1) the fusion of improved A* and DWA algorithms compered to DWA algorithm, effectively reduces the path length, planning time and increased flight speed ,also improves both the path smoothness and safety of the local obstacle avoidance and keep the local path more similar to the initially planned global path.

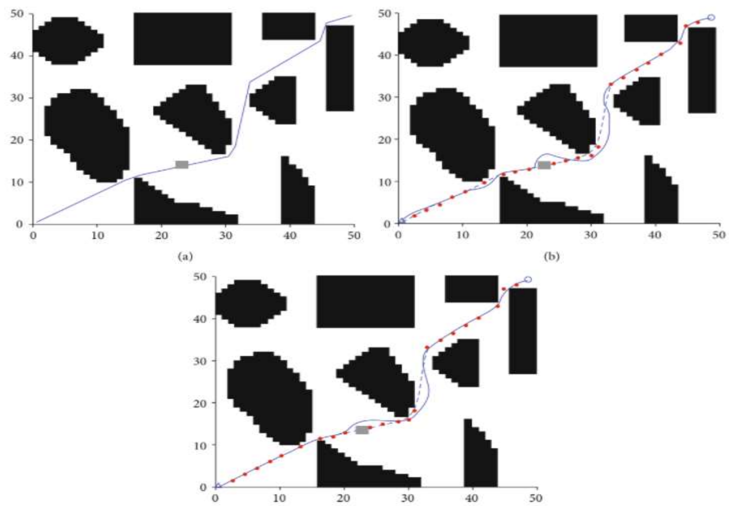


Figure 1: Comparison of the path planning algorithms on an unknown map. (a) Improved A∗ algorithm. (b) Standard DWA algorithm.(c) Fusion of improved A∗ and DWA algorithm.

In [2] Zheng and his partners present an overview of studies on UAV path planning based on CI (Computational Intelligence) algorithms , such as: Genetic Algorithm(GA), Particle Swarm Optimization (PSO),Ant Colony Optimization and others, and their contribution to the problem of UAV path planning between the years 2008-2017. Collected articles were classified according to three different aspects: algorithms ,time domain and space domain. Algorithm classification classify the articles according to the algorithms used , and for each algorithm represent how it works and what improvements the studies have presented in order to further optimize the problem of UAV path planning. The algorithm that appeared most in studies between the years 2008-2017 according to Zheng is GA and present how GA improve performance as the speed and accuracy of UAV path planning by improvements on each operation of the algorithm like :designed multi objective functions which reduce execution time of solution ,use dual population with different fitness functions which improves the search ability of GA and which ensures optimism ,change in selection method and population size which anticipated that each generation better than previous and increases the possibility of finding the best solution. Time domain classification classify the articles into offline and online. Offline path planning ,which is not designed to deal with unknown obstacles and threats, use multi modal due to the fact that different groups of obstacles cause different path planning .online path planning ,where unknown obstacles will be treated after has been detected by sensors, is a multi objective planning problem can be solved by different CI algorithms like GA and others due to the ability of react to changes in the environment by replanning local paths. Space domain classification classify the algorithms into 2D (When the height of the UAV is consonant) and 3D environments, where 3D environments creates new challenges, like geometric physical and temporal constraints) that need to be addressed .

Traditional path planning methods. The methods have their respective advantages under certain conditions, but all require to establish problem models in advance, and perform not so well when environment information cannot be obtained in advance or problem models are too complex. a path planning algorithm for UAV based on Q-learning, and the training efficiency of Q learning is improved in [4]. On this basis, a simulated antagonistic environment of the typical UAV mission is established in STAGE Scenario software and the improved algorithm is tested and verified in this simulation environment. The experimental results prove that the improved algorithm is feasible and effective for UAV path planning. Overviews of STAGE Scenario is a software tool for building and simulating tactical environment. Data Acquisition from STAGE selects the necessary state and attribute information such as UAV position, target position, etc. Q-Learning Algorithm used to find the optimal action selection strategy in MDP. Q-learning is that Agent learns an action value function to maximize the cumulative rewards obtained from the environment. Q-Learning Based Path Planning Model Environmental States in solving the path planning problem we rasterizing the environment space Reward function it is important to construct a reasonable reward function in reinforcement learning. Improvements of Q-Learning we optimizes Q-learning algorithm from two aspects: action selection strategy and Q-function initialization method. Action selection strategy It is necessary to design a suitable action selection strategy. we combine €-greedy strategy with Boltzmann strategy. Qfunction initialization method using priori knowledge to initialize Q-function is one of the important means to improve the convergence speed of Q-learning algorithm Total Proposed Method Qlearning algorithm is used to plan paths for the UAV perform reconnaissance SIMULATION AND ANALYSIS we build a simulated antagonistic environment based on STAGE and map the UAV. Results and Analysis The curve of average rewards value obtained by two algorithms is shown in Fig. 6. the average reward value obtained from the environment using improved Q-learning algorithm (IQL) exceeds 100 for the first time at 5800th episode and then remains stable above 100. The above results show that the improved algorithm accelerates learning efficiency to some degree, and the success rate of planning is also improved. Optimal paths planned by two algorithms are shown in Fig. 7. The red circular area denotes the threat area and the green circular area denotes the target area. As shown in the figure, the obtained optimal paths both reach the target area successfully. According to features of UAV path planning, the reward function is designed, and the new action selection strategy and Q-function initialization method are used to improve the Q learning algorithm, the improved algorithm has a better performance. The simulation results (see Figure 2) show that the improved algorithm possesses a stronger search performance, both the learning efficiency and the success rate of planning is improved to some extent, and it is feasible and effective for UAV path planning.

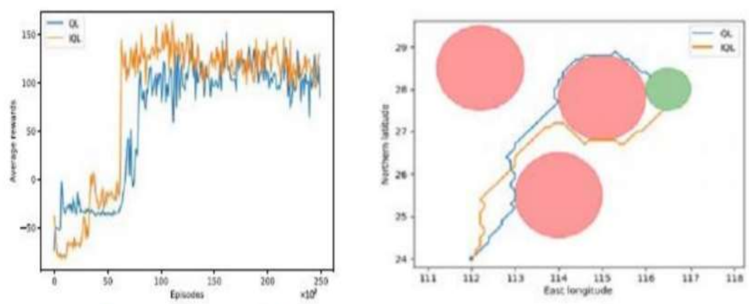


Figure 2

3.Background:

3.1 Genetic Algorithm

Genetic algorithm (GA), an optimization algorithm, which is inspired from biological evolution process is a population-based metaheuristic algorithm. GA like the Darwinian conception of the strong survival in nature. GA was proposed by J.H. Holland in 1992. The basic elements of GA are chromosome representation, fitness selection, and biological-inspired operators. Typically, the chromosomes take the binary string format. Each specific position on chromosome (gene) has two possible values - 0 and 1. Chromosomes are considered as points in the solution space. These are processed using genetic operators by iteratively replacing its population. The fitness function, which depends on the problem being investigated, is used to assign a value for all the chromosomes in the population. In selection, the chromosomes are selected based on its fitness value for further processing. The biological-inspired operators are mutation and crossover. In crossover operator, a random position in chromosome is chosen and it changes the subsequences between chromosomes (called parents chromosomes) to create off-springs (called child chromosome). In mutation, some bits of the chromosomes will be randomly flipped based on probability. The procedure of Genetic Algorithm (see Figure 3) is as follows: After solutions encoding, a population of n chromosomes are initialized randomly. The fitness of each chromosome in this population is computed. Two chromosomes selected from the population according to the fitness value. The crossover operator with crossover probability is applied on the two chromosomes selected to produce an offspring. Thereafter, mutation operator is applied on produced offspring with mutation probability to generate the new offspring, and finally the new offspring placed in new population. The selection, crossover, and mutation operations will be repeated on current population until the new population is complete.

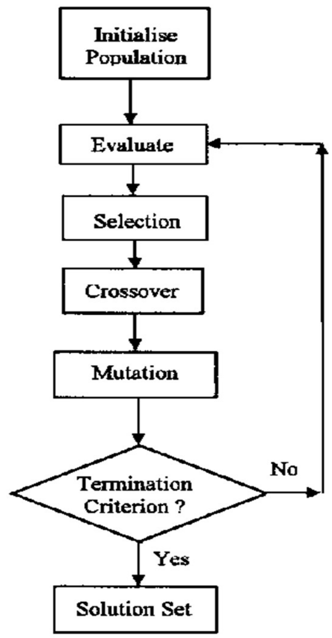


Figure 3: Genetic Algorithm steps

3.2 Genetic Algorithm Operators:

3.2.1 Encoding step:

The solutions (individuals) must be encoded in a particular bit string to be treated. The encoding types are differentiated according to the problem domain. Some of the encoding types are binary, octal, hexadecimal, permutation, value-based, and tree. Binary encoding (see Figure 4) is the most used encoding in GA. Each solution is represented as a sequence of 1 or 0, and each bit (gene) in the sequence represents the characteristics of the solution. In octal encoding, the solution is represented in the form of octal numbers (0–7). In hexadecimal encoding, the solutions are represented in the form of hexadecimal numbers (0–9, A-F). The permutation encoding is generally used in ordering problems. In this encoding type, the solution is represented by the string of numbers that represents the position in a sequence. In value encoding type, the solution is represented using sequence of some values, these values can be real, integer number, or character. It is mainly used in neural networks for finding the optimal weights. In tree encoding, the solution is represented by a tree of methods or commands. These methods and commands can be related to any programming language. This is very much like the representation of repression in tree format. This type of encoding is generally used in evolving programs or expressions. Each encoding may be more suitable for some problems than another encoding. after encoding , the individual that have been coded called chromosome. Every bit in chromosome called Allele , and a group of Alleles called Gene.

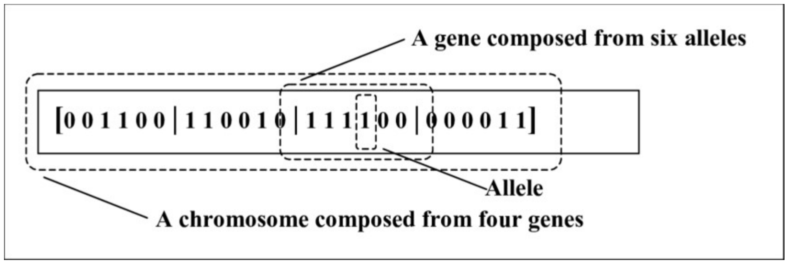
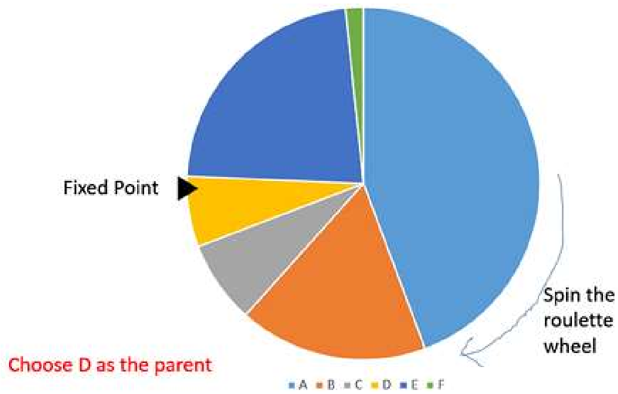


Figure 4: Binary encoding

3.2.2 Selection step:

Selection is a tool that determines which of the chromosomes after encoding will participate in the next reproduction process by selection them according to fitness function or other requirements. Some of well-known selection techniques are roulette wheel, rank, tournament and stochastic universal sampling. Roulette wheel (see Figure 5) selection maps all the chromosomes onto a wheel with a portion of the wheel allocated to chromosomes according to their fitness value. This wheel is then rotated randomly to select specific chromosomes that will participate in formation of the next generation. Rank selection is the modified form of Roulette wheel selection. It utilizes the ranks instead of fitness value. Ranks are given to chromosomes according to their fitness value so that each chromosome gets a chance of getting selected according to their ranks. Tournament selection, whereas the chromosomes are selected according to their fitness values from a stochastic roulette wheel in pairs. After selection, the chromosomes with higher fitness value are added to the group of next generation. In this selection type, each chromosome is compared with all n-1 other individuals if it reaches the final population of solutions. Stochastic universal sampling (SUS), which is an extension to the existing roulette wheel, it uses a random starting point in the list of chromosomes from a generation and selects the new individual at evenly spaced intervals. It gives equal chance to all the chromosomes in getting selected for participating in crossover for the next generation. Elitism selection was proposed for improving the performance of Roulette wheel. It ensures that the best fitness chromosomes in a generation is always propagated to the next generation by add them to the next generation if they didn't select. Each selection type may be more suitable for some problems according to the population size decided to select.



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Figure 5: Roulette wheel selection

3.2.3 Crossover step:

Crossover operators are used to produce the new offspring chromosomes by combining the genetic information of two or more parents chromosomes. Some of well-known crossover operators are single- point, two-point, k-point, uniform, partially matched, order, precedence preserving crossover, shuffle, reduced surrogate and cycle crossover. In a single point crossover, the crossover point is selected randomly. The genetic information of two parents chromosomes which is beyond that crossover point will be swapped with each other to get the new offspring chromosome. In a two point (see Figure 6) and k-point crossover, two or more crossover points are selected randomly and the genetic information of parents chromosomes will be swapped as per the segments that have been created. In a uniform crossover, parent cannot be decomposed into segments. The parent can be treated as each gene separately. We randomly decide whether we need to swap the gene with the same location of another chromosome. Partially matched crossover (PMX), whereas two parents are choose for mating. One parent donates a portion of the genetic material and the corresponding portion from the other father shares the child. Once this process is complete, the alleles left are transcribed from the second parent. The order crossover (OX) copies one (or more) part of the original to the offspring chromosome from the selected cut points and fills the remaining space with values other than those included in the transcribed section. Shuffle crossover shuffles the values of an parent chromosome before the crossover and unshuffled them after crossover operation is performed so that the crossover point does not introduce any bias in crossover. Cycle crossover attempts to generate an offspring chromosome using parents chromosomes where each element occupies the position by referring to the position of their parents. In the first cycle, it takes some elements from the first parent chromosome. In the second cycle, it takes the remaining elements from the second parent chromosome.

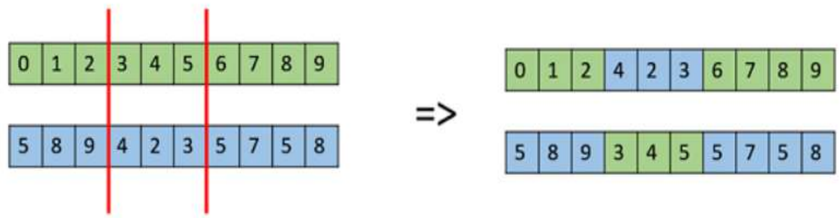


Figure 6: two-point crossover

3.2.4 Mutation step:

Mutation operators are used to produce new offspring chromosomes by mutation the genetic information of new offspring produced in crossover step. Some of well-known mutation operators (see Figure 7) are displacement, simple inversion, and scramble mutation. Displacement mutation (DM) operator displaces a substring of a given chromosome within itself. The place is randomly chosen from the given substring for displacement such that the resulting solution is valid as well as a random displacement mutation. In Exchange mutation and insertion mutation operators, a part of a chromosome is either exchanged with another part or inserted in another location. The simple inversion mutation operator (SIM) reverses the substring between any two specified locations in a chromosome. SIM is an inversion operator that reverses the randomly selected string and places it at a random location. The scramble mutation (SM) operator places the elements in a specified range of the chromosome in a random order and checks whether the fitness value of the recently generated solution is improved or not.

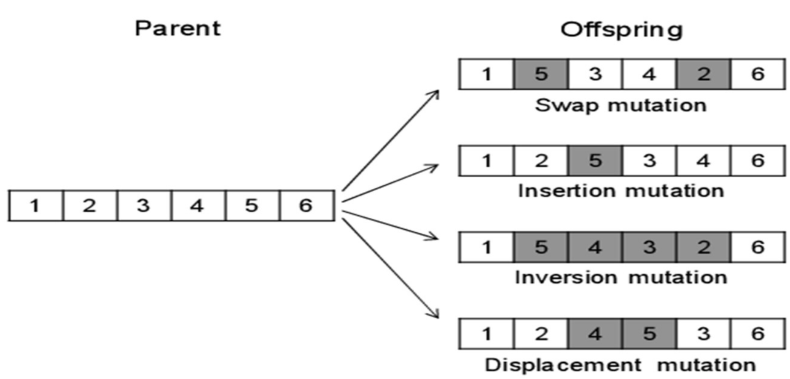


Figure 7: Mutation operators

4.Expected Achievements:

Based on the genetic algorithm, the various mutations allowed by the structure of the algorithm and his steps, We expect that the solution we have proposed will achieve two main objectives:

1- correct and accurate paths for UAVs that cover all the targets.
2- reduce the iterations number of the two sub algorithms we proposed compared with the two sub algorithms proposed by Lin In[5].

5.Research / Engineering Process: Process:

In order to solve the problem of Multi-UAV path planning, we will use a Genetic Algorithm (GA) and K-means clustering algorithm based on GA. The solution depends on the fact that the initial locations of the UAVs takeoffs and maritime targets we would like to extract and reach them at minimal iterations of computing path for each UAV which can also reduce total time of the rescue mission are given, We therefore create random suitable initial locations for each UAV and target. The solution consists of two sub algorithms: First Sub-algorithm of marine search and rescue targets classification based on K-means and genetic algorithm, and second sub-algorithm for maritime targets path planning based on genetic algorithm and mean-variance. The purpose of the solution we propose here to reduce the time and the iterations amount of path planning for multi-UAV based on genetic algorithm, compared with the solution proposed by Lin In[5].

First Sub-algorithm: K-means clustering algorithm based on Genetic Algorithm:

By K-means clustering algorithm cluster the set of targets into clusters, where the clusters number is equal to UAVs number:

a- Encoding:
   Using the positions of the cluster centers to represent the chromosome. if $(x1, y1)$,$(x2, y2)$, ..,$(xm,\ ym)$ are the cluster centers ((x, y) are latitude and longitude coordinate position), the chromosome code will be: ($x1$, $y1$, $x2$, $y2$, …, xm, ym).
b- Fitness Function:
   Use criterion function of K-means to construct fitness function, which reflects the accuracy of the clusters, to make fitness function for evaluated each chromosome Separately. The larger the fitness function, the better the clustering result:

$$\max: f = \frac{1}{1 + E}$$

   E - Efficiency of the K-means algorithm

c- Selection:
Using two strategies for selection : A half of the new population will consist of the chromosomes with the best fitness values, and selection the other half of the new population selection by the tradition roulette wheel. In this approach we try to keep the good solutions in the hope that a change in them by the operators of the genetic algorithm will further improve them, while giving opportunity to the rest of the solutions to be selected and perhaps by the changes and mutations even get the best solution from them.

d- Crossover:
We select an adapted two-point crossover method, which suitable for the chromosome structure, choosing two random points and swap them with the suitable points in the other chromosome (swapping cluster center points).

e- Mutation:
Using the uniform mutation operator to select the mutation point, generate random numbers from the value range, and replace the current value by the new.

f- Run K-means clustering algorithm with the new population, evaluate every chromosome and back to step c if don't reach the number of cluster iterations .

Second sub-algorithm: Genetic Algorithm based on mean-variance for path planning:

After clustering computed, here we propose a multi-chromosome genetic algorithm based on the mean-variance model (as the fitness function) to compute the UAVs paths, using multi-chromosome optimization search, instead of the traditional method of evaluating each chromosome independently here we look at the average fitness value of several chromosomes (which together cover all targets), in order to achieve global optimization.

a- Multi-chromosome coding:
Each gene on the chromosome represents a target on the path of current UAV this chromosome represents, where the first gene present the flight starting point, the second gene is the next target, every chromosome may have different length which depending on the targets assigned by GA operations for the UAV that this chromosome represents and each group of chromosomes (multi-chromosome), the number of chromosomes in a group as the number of UAVs, represents paths for these UAVs respectively.

b- Fitness function based on the mean-variance model:
The mean (average) value of the flying distances is taken as the primary objective function, the smaller the mean is the better, and the variance of the flying distance of the UAV is selected as the secondary objective function, where less variance is the better. This approach makes it possible to achieve

an optimal global solution for multi-UAV path planning accordingly to the length of all the routes of the UAVs.

$$min\, f(\mu, \sigma^2)$$

among,

$$\begin{cases} \mu = \dfrac{\sum_{i=0}^{n}\sum_{j=0}^{n} s_{ij} \times \sum_{k=1}^{m} r_{ijk} \times w_{ijk}}{m} \\ \sigma^2 = \dfrac{\sum_{k=1}^{m}(S_k - \mu)^2}{m} \end{cases}$$

C = {1,2, …, n}: targets set, the first target is the starting point of the flight.
U = {1,2, …, m}: collection of UAV.
S_{ij}: path length of the UAV from target i to target j.
S_k: total path length of the k-th UAV.
r_{ijk}: the k-th UAV flies from target i to target j.
w_{ijk}: weight of flying from target i to target j from the k-th UAV.
μ: mean value of the flying distance of the UAVs.
σ^2: variance value of the flying distance of the UAVs.

c- Selection: Using rank selection strategy. Ranks are given to chromosomes according to their fitness value and then by the roulette strategy select chromosomes from each rank. This approach allows every chromosome of any rank to be selected.

d- Crossover:
choosing randomly two different genes, swap these gens places and getting the new offspring of current UAV. The goal of this operator to achieve a local optimal path.

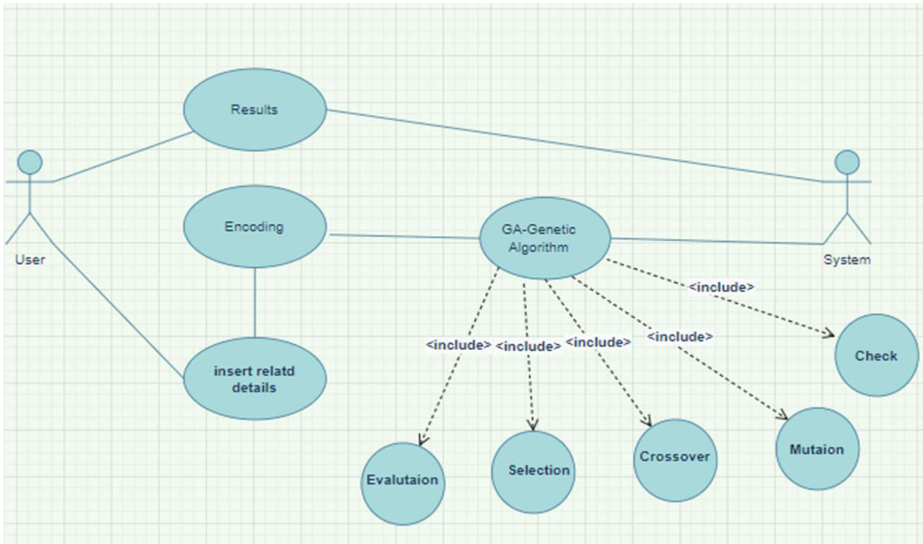
e- Mutation:
Using gene segment insertion and swap mutation. When the target position on the two chromosomes is close, the target gene segment is marked as the exchange region gene, which obtaining the global optimal solution. If the fitness value after mutation is less than before mutation, perform mutation operation. Also using the swap mutation by random two gene positions and swap the values in order to achieve local optimization.

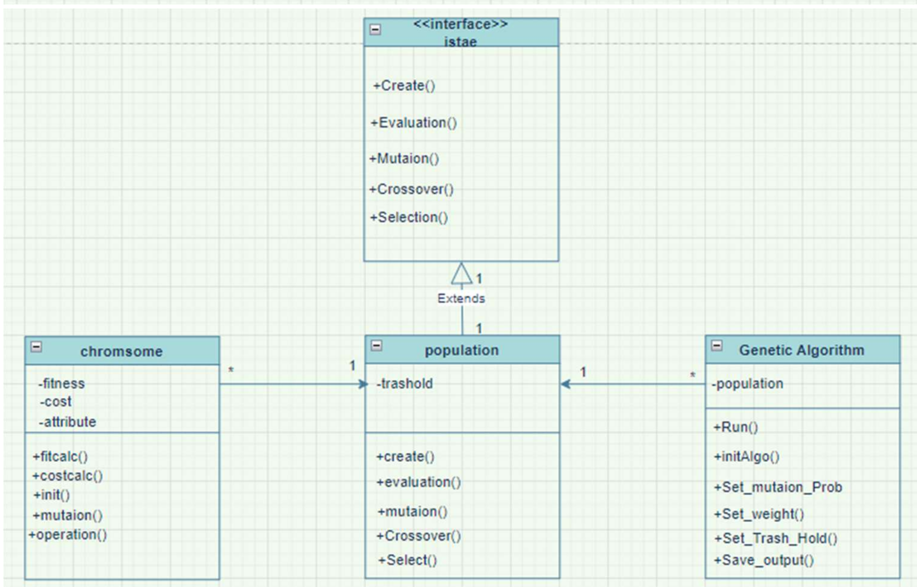
f- Evaluate the new population and back to step c.

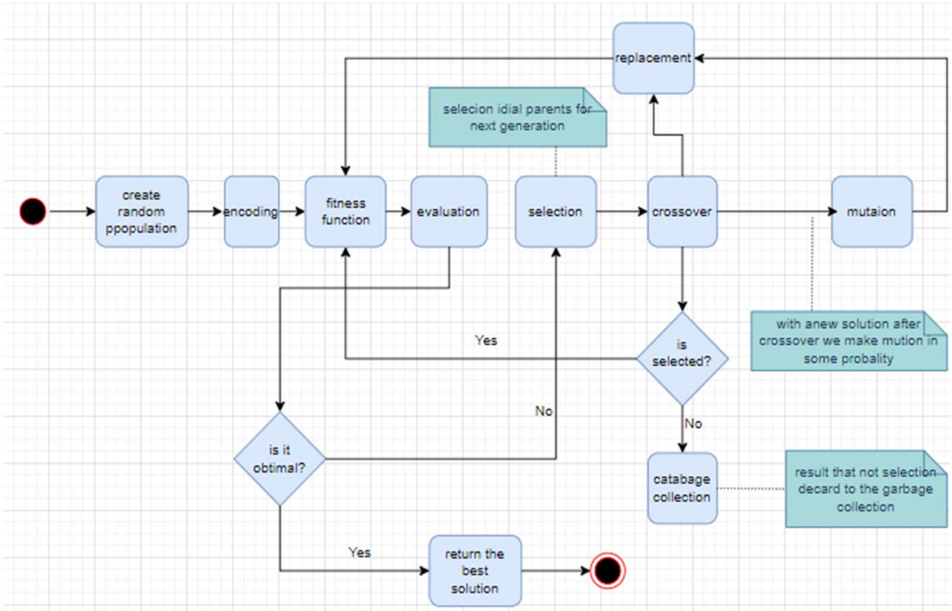
The two sub-algorithms presented above make up the solution algorithm we propose which relies on the ability of the operators of the genetic algorithm to create and find the optimal solution. In order to test, evaluation and simulate we plan to implement the proposed algorithm in java language with a suitable UI by testing different conditions, such as a change in the number of UAVs and the number of targets, how many iterations will be performed until the optimal solution is achieved.

6.Product:

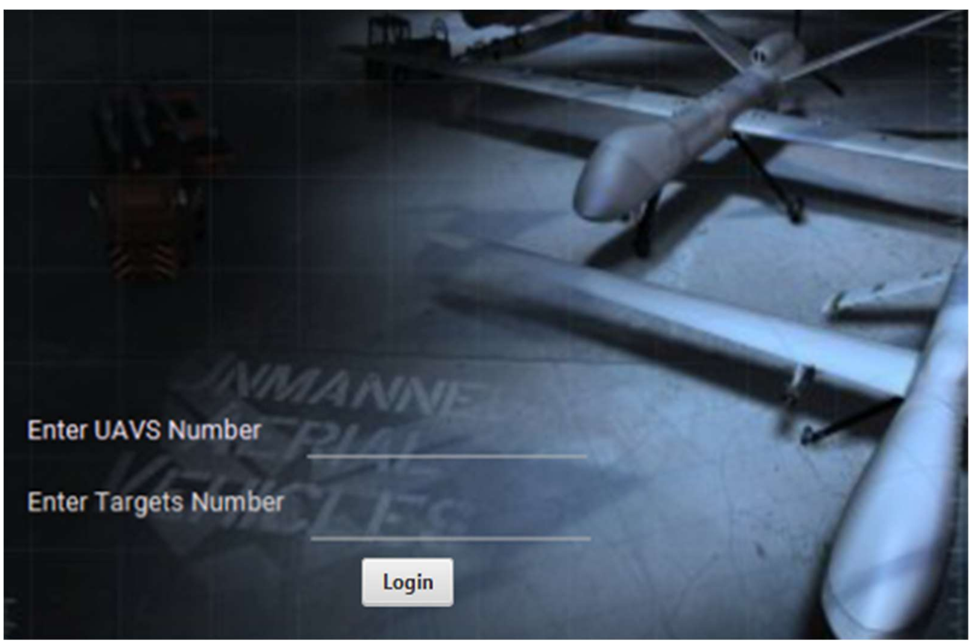
6.1 DESIGN – UML DIAGRAMS

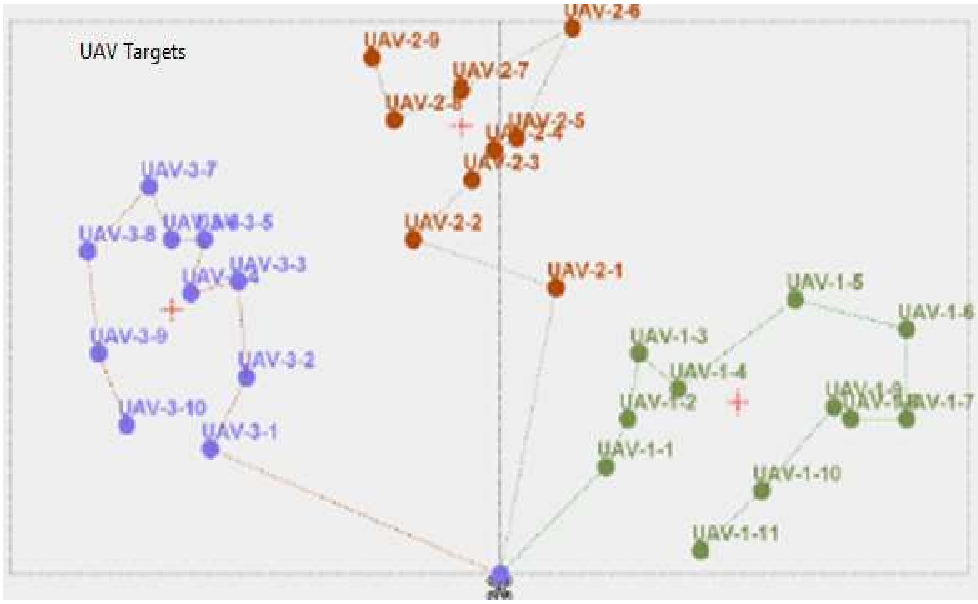


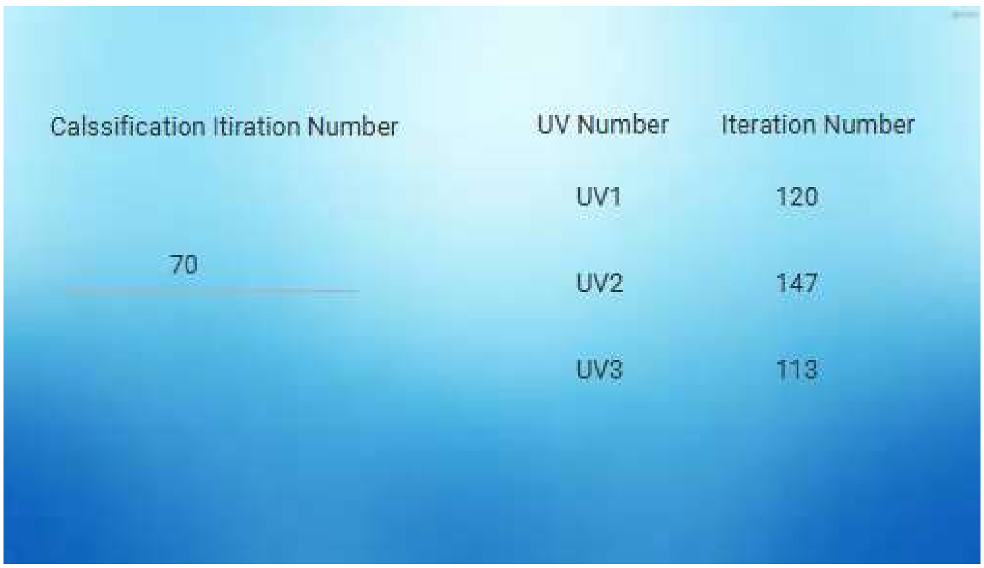



replacement
selecion idial parents for next generation
create random ppopulation
encoding
fitness function
evaluation
selection
crossover
mutaion
with anew solution after crossover we make mution in some probality
Yes
is selected?
No
is it obtimal?
No
catabage collection
result that not selection decard to the garbage collection
Yes
return the best solution

6.2 GUI:


Enter UAVS Number
Enter Targets Number
Login


UAV Targets
UAV-2-6
UAV-2-9
UAV-2-7
UAV-2-8
UAV-2-5
UAV-2-4
UAV-2-3
UAV-2-2
UAV-2-1
UAV-3-7
UAV-3-8
UAV-3-5
UAV-3-3
UAV-3-9
UAV-3-2
UAV-3-10
UAV-3-1
UAV-1-5
UAV-1-6
UAV-1-3
UAV-1-4
UAV-1-2
UAV-1-7
UAV-1-1
UAV-1-10
UAV-1-11


Calssification Itiration Number
70
UV Number
Iteration Number
UV1
120
UV2
147
UV3
113

7. Evaluation/Verification Plan:

The table below shows the unit test we will perform for each class and what will be tested.

Test	Test name	Test Description
Locations	Checking location values	We will check that the function that calculated initial random locations, for targets and starting positions for UAVs, creates appropriate values.
Population (targets array)	Get a specific chromosome	During the algorithm we need to choose a specific chromosome, this test makes sure that it will choose the correct one.
	Fitness probability for roulette	The roulette selection function relies on the pie created by the probabilities of the chromosomes.
	Get and set population	During the algorithm, we need the population for many calculations, for example, compute a path length. We want to make sure this function is fully operative and that the population cannot be changed during a specific run.
	Get the best solution out of the population (single and multi-chromosome)	For the output result or algorithm steps, we will check that the best solution got.
Genome	Checks the solution evaluation	In order to check how good the solution is, we need to evaluate it. We therefore check the fitness.
	Check the chromosome range in the roulette	The ranges are float numbers between 0-1.
	Test the chromosome sequence	We tested the sequence length and that there are no repeats, meaning that a target will not appear twice in a sequence.
Utility	Test attributes boundaries	Once we got an input that includes a values represent the UAVs and targets number, we need to make sure that the user inserted the values within the boundaries, for example, targets number can't be -10 .

	Checks the input	In order to run the algorithm we need to make sure that the input exists and contains values of UAVs and targets number.

8. REFERENCES:

1-Hu, J.; Lanzon, A."An innovative tri-rotor drone and associated distributed aerial drone swarm control". Robotics and Autonomous Systems.

2-Yijing Zhao, Zheng Zheng, Yang Liu. Survey on computational-intelligence-based UAV path planning.

3-Xiong Bai, Haikun Jiang, Junjie Cui, Kuan Lu, Pengyun Chen , and Ming Zhang. UAV Path Planning Based on Improved A∗ and DWA Algorithms.

4-Chao Yan, Xiang Xiaojia. A Path Planning Algorithm for UAV Based on Improved Q-Learning.

5-Lin Li, Qun Gu, Li Liu. “Research on Path Planning Algorithm for Multi-UAV Maritime Targets Search Based on Genetic Algorithm”.

6-Tom V. Mathew Indian Institute of Technology Bombay, Mumbai-400076.” Genetic Algorithm”.

7-Sourabh Katoch1 & Sumit Singh Chauhan1 & Vijay Kumar. “A review on genetic algorithm: past, present, and future”.

8- G.E. LIEPINS and M.R. HILLIARD. “GENETIC ALGORITHMS: FOUNDATIONS AND APPLICATIONS”.

9- Abdurrahim Sonmez, Emre Kocyigit and Emin Kugu. “Optimal Path Planning for UAVs Using Genetic Algorithm”.

10- Shubhani Aggarwal, Neeraj Kumar. “Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges”.

11- Halit Ergezer · Kemal Leblebicioglu. “3D Path Planning for Multiple UAVs for Maximum Information Collection”.