Navigation and Guidance Planning For Air Vehicles

Ferhat Uçan [a,b], D. Turgay ALTILAR [b]

[a] *TUBITAK Marmara Research Center Information Technologies Institute*
41470, Gebze, Kocaeli
[b] *Istanbul Technical University, Computer Engineering Department,*
34469, Maslak, İstanbul
ucanf@itu.edu.tr, altilar@itu.edu.tr

Abstract

Path planning is the art of deciding which route to take, based on and expressed in terms of the current internal representation of the terrain. Path finding also involves the execution of this theoretical route, by translating the plan from the internal representation in terms of physical movement in the environment.

During combat, transfers of military units in the shortest period and with least damage are vitally important. In this study, navigation planning problems, described in semi-dynamic and fully-dynamic scenarios, have been solved using evolutionary algorithm approach. 3-D graph is used to mode the environment. In the first scenario, graph node and edge deletion and cost update operations are applied and the impact of the different parameters of genetic algorithm on the solution has been investigated. In the second scenario, in an environment with possibility of combat, the aim is to find the most secure shortest path by employing genetic algorithm approach dynamically, in case any combat information arrives during the transfer of units.

1. Introduction

During combat, transfers of military units in the shortest period and with least damage are vitally important. In this study, path planning problem is considered for flight planning and navigation planning. Navigation planning problems, described in different scenarios, have been solved using evolutionary algorithm approach. The aim is to prepare the path for the flight plan and guide the pilot of the air vehicle by online execution of the flight plan.

Prior to departure, each commercial aircraft must have a flight plan from its origin to its destination [1,2]. Flight plan is a flight path followed by pilot prior to departure. Flight plans include basic information such as departure and arrival waypoints, estimated time to arrive to the next point, alternate airports, flight type, waypoints and fuel information. It describes the route, the altitudes, and the speeds which the plane is expected to fly for the trip, and it affects fuel consumption, safety, speed, arrival time, and passenger comfort. In most countries, flight plans are required for flights under instrument flight rules, they are optional unless crossing national borders, however they are highly recommended, especially when flying over inhospitable areas, such as water, as they provide a way of alerting rescuers if the flight is overdue. The flight plan includes air vehicle motion modeling fundamentals that are important parts of a tactical environment simulation software. Pilot may prepare flight plans by choosing from pre-defined mark points and play a simulation scenario.

Navigation and routing problems are very general problems. They can be used for vehicle routing in traffic, in military applications, in robotics, for determining the route between two points in urban transportation [3,4,5]. or for routing data packets in a network. In recent years, the optimization became more important due to the possibility of solving many large combinatorial optimization problems and multi-objective engineering problems. In this paper, the dynamic path planning problem is solved with evolutionary methods, for compensating the existing deficiencies of the other approaches.

2. Evolutionary computing

Evolutionary algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved. Algorithm is started with a set of solutions called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions

IEEE
computer
society

which are selected to form offspring are selected according to their fitness values. The more suitable they are, the more chances they have to reproduce. This is repeated until some condition. Stopping criteria may be number of iterations, or individual similarity or other criteria [6].

There are many versions of genetic algorithms, one differing from another in some detail. In a nutshell, all genetic algorithms have two basic steps: during the selection step, a decision is made as to who in the population is allowed to produce offspring, and during the replacement step another decision is made as to which of the members from one generation are forced to perish in order to make room for an offspring to compete [7,8]. The random makeup of the initial population has a large effect on the likelihood that evolutionary programming will find a successful program. The main flow of a genetic algorithm is shown in Figure 1.
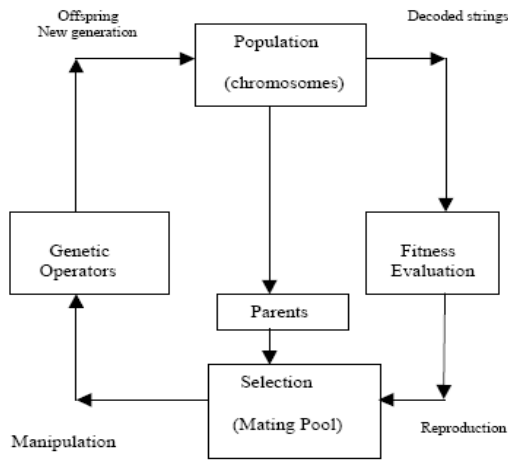

Figure 1 The flow of a basic genetic algorithm

3. Navigation and guidance calculations

Shortest distance between two points is a straight line. However anyone attempting to fly from Los Angeles to New York on the straight line connecting them would have to dig a very substantial tunnel first. The shortest distance, following the earth's surface lies vertically above the aforementioned straight line route. This route can be constructed by slicing the earth in half with an imaginary plane. This plane cuts the earth in a circular arc connecting the two points, called a great circle. Only planes through the center of the earth give rise to great circles. Any plane will cut a sphere in a circle, but the resulting little circles are not the shortest distance between the points they connect. A little thought will show that lines of longitude (meridians) are great circles, but lines of latitude, with the exception of the equator, are not.

Natural questions are to seek the great circle distance between two specified points and true course at points along the route. The required spherical trigonometric formulae are greatly simplified if angles and distances are measured in the appropriate natural units, which are both radians! A radian, by definition, is the angle subtended by a circular arc of unit length and unit radius. Since the length of a complete circular arc of unit radius is 2*pi, the conversion is 360 degrees equals 2*pi radians. The nautical mile is currently defined to be 1852 meters - which to be consistent with its historical definition implies the earth's radius to be 1.852 * (180*60/pi) = 6366.71 km, which indeed lies between the currently accepted (WGS84) equatorial and polar radii of 6378.137 and 6356.752 km, respectively. Other choices of the earth's radius in this range are consistent with the spherical approximation and may for some specialized purposes be preferred.

The general functions used for navigation planning are distance, bearing, position and intersection point calculations. In order to calculate the great circle distance and bearing angle between two waypoints the following equations are used;

$$\mathbf{P_1} = \cos(L_1)\cos(\lambda_1)\mathbf{i} + \cos(L_1)\sin(\lambda_1)\mathbf{j} + \sin(L_1)\mathbf{k}$$

$$\mathbf{P_2} = \cos(L_2)\cos(\lambda_2)\mathbf{i} + \cos(L_2)\sin(\lambda_2)\mathbf{j} + \sin(L_2)\mathbf{k}$$

$$\text{dist12} = \tan^{-1}\left(\frac{|\mathbf{P_1} \times \mathbf{P_2}|}{\mathbf{P_1} \cdot \mathbf{P_2}}\right)$$

$$\boldsymbol{\eta}_{P1P2} = \frac{\mathbf{P_1} \times \mathbf{P_2}}{|\mathbf{P_1} \times \mathbf{P_2}|}, \quad \boldsymbol{\eta}_{P2P1} = \frac{\mathbf{P_2} \times \mathbf{P_1}}{|\mathbf{P_2} \times \mathbf{P_1}|}$$

$$\psi_{12} = \tan^{-1}\left(\frac{-\eta_{P1P2}(3)}{P_1(1)\eta_{P1P2}(2) - P_1(2)\eta_{P1P2}(1)}\right)$$

$$\psi_{21} = \tan^{-1}\left(\frac{-\eta_{P2P1}(3)}{P_2(1)\eta_{P2P1}(2) - P_2(2)\eta_{P2P1}(1)}\right)$$

a - origin   b - destination
D = angular distance along path
C = true bearing from the origin to the destination measured from north.
D = acos (sin(lata)·sin(latb)+cos(lata)·cos(latb)·cos(lonb - lona))
C = acos ([sin(latb) - sin(lata)·cos(D)] / [cos(lata)·sin(D)])

Given a start point, initial bearing, and distance, in order to calculate the destination point coordinates, the following formula is used;

lat = asin(sin(lat1)*cos(D)+cos(lat1)*sin(D)*cos(C))
f1 = sin(C)*sin(D)*cos(lat1)
f2 = cos(D)-sin(lat1)*sin(lat)
lon=mod(lon1-atan2(f1,f2)+pi,2*pi)-pi

In order to find the intersection of two great circles defined by the arcs from pt1 to pt2 and from pt3 to pt4, normalization vector and unit vectors are used.
e ={ex,ey,ez}={cos(lat)*cos(lon), -cos(lat)*sin(lon), sin(lat)}

lat=atan2(ez, sqrt(ex^2 + ey^2)) lon=atan2(-ey, ex)

P(e1,e2) =(e1 X e2)/‖e1 X e2‖

{e1y *e2z -e2y *e1z, e1z *e2x -e2z *e1x, e1x *e2y -e1y *e2x}

‖e‖ =sqrt(ex^2 + ey^2 + ez^2)

e1Xe2={sin(lat1-lat2)*sin((lon1+lon2)/2)*cos((lon1lon2)/2)-sin(lat1+lat2) *cos((lon1+lon2)/2) *sin((lon1-lon2)/2) , sin(lat1-lat2) *cos((lon1+lon2)/2) *cos((lon1-lon2)/2) + sin(lat1+lat2) *sin((lon1+lon2)/2) *sin((lon1-lon2)/2) , cos(lat1)*cos(lat2)*sin(lon1-lon2)}

ea= (e1 X e2)/ ‖e1 X e2‖ , eb=(e3 x e4)/‖e3 X e4‖

invert ea X eb.

two candidate intersections → (lat,lon), (-lat,lon+pi).

For the simulations of the proposed algorithm, matlab and simulink are used. Flight plan and air navigation process of the air vehicles from a departure waypoint to a descent waypoint is simulated. The simulation environment used to verify the navigation algorithms is shown in Figure 2. The simulator represents a simulation of the aircraft in open-loop flight. That is, all aircraft control inputs are set to fixed values, independent of the aircraft states. The lateral dynamics of the aircraft is stabilized by adding a wing leveler. This is implemented using proportional and integral feedback from bank angle to ailerons.
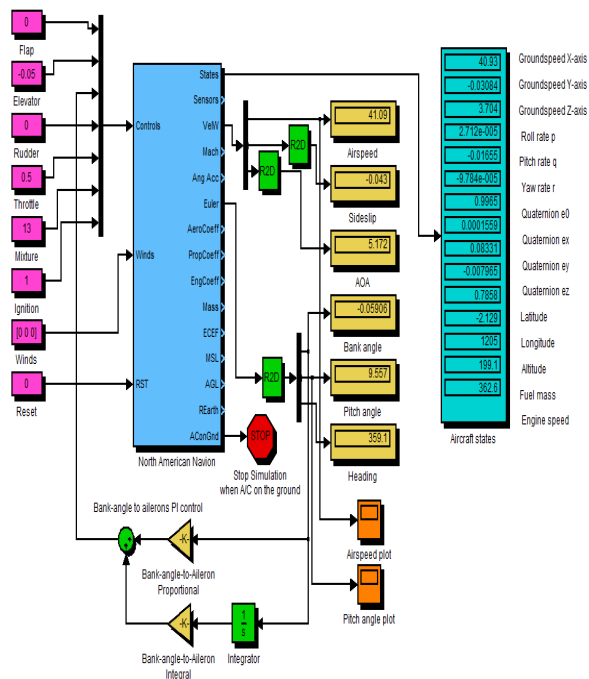

Figure 2 The simulation envirınment

Some navigation algorithms are used for the guidance of air vehicles. The flight plan execution command calculations used to guide for lateral and vertical navigation are shown below;

Dist=distance(fromWP,toWP)

Bearing1=bearing(fromWp,toWP)

Bearing2=bearing(toWP,toPlusOneWP)

deltaY=asin(sin(Dist)*sin(Bearing1-Bearing2))

deltaX=heading_angle-Bearing2

tmp1=deltaY*gravity acc/ground_speed

bank=atan(tmp1-deltaX)*(ground_speed/gravity acc)

track=atan(groundspeed_earth/groundspeed_north)

alt_dif=(altitude-altitudeToWP)

tmp2=alt_dif* gravity acc/verticalSpeed

tmp3=sqrt(alt_dif^2+Dist^2)

pitch_com=(tmp3-tmp2)*(verticalSpeed/gravityacc)

tmp4=0.5*FreeAirTemp-sqrt(altitude)

tmp5=17*sqrt(trueAirSpeed)+7*(gross_weight)

fuel_flow_rate=tmp4+tmp5-150

4. Planning flight route

Genetic algorithms imitate the evolutionary process in order to solve the optimization problems. Instead of developing one solution candidate, genetic algorithms form a set of individuals. The set, which contains probable solution candidates, is defined as population in genetic algorithm terminology. Population occurs from arrays called vector, chromosome or individual. The each element of an individual is called gene. Variable length chromosomes are used in dynamic path planning systems in order to cover the whole search space [9]. In evolutionary programming method, individuals in the population are determined by the operators of the evolutionary algorithm. In the problems like path planning, the permutation representation is used and the operators differ from the operators of the basic genetic algorithm [10]. The most important parameters of the genetic algorithm are crossover rate, mutation rate and the number of individuals in a population. In order to declare crossover and mutation rates, different values are tested and by this way the most appropriate values are found for these parameters. Chromosome number in the population is determined according to the nodes of the graph topology.

In the proposed heuristic approach, in order to represent the routes, variable-length chromosomes are used. Chromosomes are encoded by permutation encoding method. Each gene of a chromosome represents a node in the graph, and all genes show the whole path in the graph. The quality of the individuals are determined by the fitness function. A general fitness function is developed in order to meet all the constraints. Chromosomes are arranged in decreasing order according to the fitness values. In order to increase the quality of the population, a selection operator, that rise the chance of the better fit individuals, is used in the proposed algorithm. Selection operator, forces to search the solution in the

determined locations of the search space. In the proposed algorithm, in order to save the better-fit individuals for the next generation, and in order to avoid from statistical errors caused by sampling, the roulette wheel selection technique is used for selection. According to the roulette wheel selection, the fitness values of all individuals are added. The selection probability of an individual is found by dividing the fitness value to the overall value. In the crossover phase, the genes after the crossover site are exchanged between parent chromosomes. Probable crossover sites are the regions where the genes of the parent chromosomes are identical. If the parents have no identical gene pair, the crossover operator can't be applied. If there are more than one identical gene pairs, one of the pairs is chosen randomly. Crossover may generate infeasible chromosomes that violate the loop constraint and may form individual chromosomes with cycles. It must be noted that none of the chromosomes of the initial population or after the mutation is infeasible because when once a node is chosen, it is excluded from the candidate nodes forming the rest of the path. Algorithm makes a post-processing operation and removes the cycles from the individual. The sample crossover and repair operations are shown in the Figure 3.
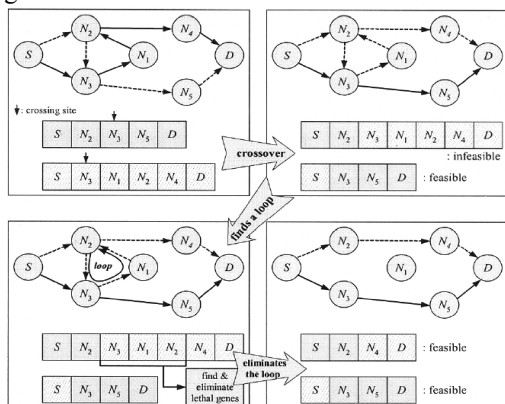
**Figure 3** Crossover and repair operators

Mutation operator increases the variation in the population, and avoids attach local optima by changing the genes of the potential chromosome. In the route planning problem, a random gene change may generate an infeasible chromosome, so two-point mutation is applied and the genes in the region between the mutation points are modified with a different route. A sample mutation operation is shown in the Figure 4.
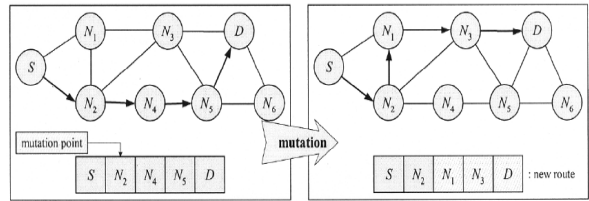
Figure 4. Mutation operator

Proposed genetic algorithm continues to operate the reproduction phase until the stopping criteria satisfied. The stopping criteria chosen for this kind of problem is individual similarity.

The runtime performance and the complexity of the proposed algorithm is compared with deterministic methods. When segment insertion/deletion/cost update operations occur, the proposed algorithm doesn't start calculations from scratch, and it converges to the solution in a shorter time than Dijkstra. Figure 5 shows the performance of the proposed algorithm at each dynamic change. The results support the view that evolutionary algorithms are an effective, robust search procedure for NP complete problems in the sense that, although they may not outperform highly tuned, problem-specific algorithms, Evolutionary algorithms can be easily applied to a broad range of NP-complete problems with performance characteristics no worse than the theoretical lower bound of an N^3 speedup.
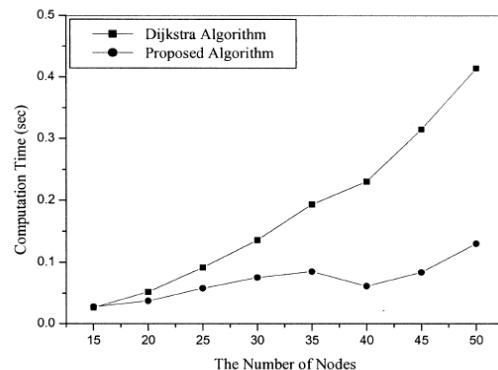
Figure 5. Run-Time Performance

The most recent dynamic path planning solution techniques are Ramalingam and Reps [11], Franciosa et al. [12], and Frigioni et al. [13]. The solution by Franciosa et al. can be used only for semi-dynamic case. The Ramalingam and Reps' solution was found is successful concerning run-time, Frigioni et al.'s is better when the number of segments to be updated had to be minimized. The existing fully dynamic algorithms process unit changes to topology one modification at a time, but when there are several such operations occurring in the environment simultaneously, the algorithms are quite inefficient. The problems are worse in large topologies which have

a large number of nodes and edges, where a large number of topology modifications occur continuously at all times. In such cases, the existing algorithms may fail to determine the shortest path information in a time critical manner. The proposed algorithm may respond to the concurrent weight updates in a shorter time for dynamic environments. Figure 7 shows the higher performance of the proposed algorithm especially for strongly connected graphs. When the number of nodes increases, the computation time difference between the Dijkstra algorithm and the proposed algorithm in dynamic scheme also increases. As evolutionary algorithms are stochastic and heuristic methods, the computation time statistics of the genetic algorithm are the averages of 30 simulations.

5. Conclusion

Deterministic algorithms like Floyd, Dijkstra or heuristic methods like neural network and A-Star may be used for the solution of navigation planning problem in static schemes. Dynamic path planning algorithms like Frigioni, Franciosa, Ramalingam Reps may be used in dynamic environments but they are constrained by some limitations. When several concurrent changes occur in the environment simultaneously, these algorithms are quite inefficient. When the segment costs and conditions change stochastically and continuously, these algorithms fail to converge the actual underlying average solution. Also the algorithm should work with uncertain graphs.

The proposed algorithm doesn't use a graph matrix, owing to the reproduction loop, selection mechanism and the fitness function, the algorithm finds results in a shorter computation time compared with the analytical algorithms. The proposed algorithm provides advanced search speed, quality and flexibility in dynamic schemes. In addition to the flight planning algorithm, a flight execution algorithm is developed to guide the pilot follow the mission plan. The execution algorithm provides flight commands like pitch, bearing, track and bank angles, desired ground speed to arrive next point on time.

6. References

[1] Cohen, M. S., Leddo, J. H., and Tolcott, M. A. (1989). Personalized and prescriptive aids for commercial air flight replanning (Report No. 89-2). Reston, VA: Decision Science Consortium, Inc.

[2] Sorensen, J. A., Waters, M. H., and Patmore, L. C. (1983). Computer programs for generation and evaluation of near-optimum vertical flight profiles LContractor Report No. 3688). Hampton, VA: NASA.

[3] Munemoto M., Takai Y., Sato Y., "An Intelligent Network Routing Algorithm by a Genetic Algorithm", Proceedings of the Fourth International Conference on Neural Information Processing, pp.547-550, 1997

[4] Siregar J.H., Zhang Y., Takagi H.," Optimal Multicast Routing Using Genetic Algorithm for WDM Optical Networks", IEICE Tran. Commun, Vol.E88–B, No.1, 2005

[5] Baluja, S. (1996) "Evolution of an Artificial Neural Network Based Autonomous Land Vehicle Controller," IEEE Transactions on Systems, Man and Cybernetics, Vol. 26 No. 3, June 1996.

[6] Goldberg G,D.E.,1989.Genetic Algorithms in Search Optimization and Machine Learning, Addion Wesley Publishing Company,USA.

[7] Jonathan Shapiro Genetic Algorithms in Machine Learning Volume 2049, Issue , pp 0146-Lecture Notes in Computer Science.

[8] Chambers, L.D. (1999), Practical Handbook of Genetic Algorithms: Complex Coding Systems Volume III, CRC Press.

[9] Harik G., Cantu-Paz E., Goldberg D. E., Miller B. L., "The Gambler's ruin problem, genetic algorithms, and the sizing of populations", Evol. Comput., vol. 7, no. 3, pp. 231–253, 1999.

[10] Ahn C. H., Ramakhrisna R. S., "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations", IEEE Trans Evolutionary Computation, Vol.6, No.6, 566-579, 2002.

[11] Ramalingam G., Reps T., "On the computational complexity of dynamic graph problems," Theoret. Comput. Sci., vol. 158, no. 1, pp. 233–277, 1996.

[12] Franciosa P. G., Frigioni D., Giaccio R., "Semi-dynamic shortest paths and breadth first search in digraphs," in Symp. Theoretical Aspects of Computer Science, vol. 1200, Lecture Notes in Computer Science, 1997.

[13] Frigioni D., Marchetti A., Nanni U., "Fully dynamic algorithms for maintaining shortest paths trees," J. Algorithms, vol.34, pp.251–281, 2000.