A Heuristic Approach to Military Transition Problem

Ferhat Uçan, D. Turgay ALTILAR

Istanbul Technical University, Computer Engineering Department, 34469, Maslak, İstanbul

ucanf@itu.edu.tr, altilar@itu.edu.tr

Abstract

Dynamic path planning problem, which is defined by finding the shortest distance movement from a starting point to a target destination in an environment that contains dynamic moveable obstacles, is a problem that is encountered in different forms in many fields. In military field, the safest transition of military units in shortest time is one of these forms. Transition problem is a very general problem. It can be used for vehicle routing in traffic, in military applications, in robotics, for determining the route between two points in urban transportation or for routing data packets in a network.

1. Introduction

In recent years, the optimization became more important subject due to the possibility of solving many large combinatorial optimization problems and multi-objective engineering problems. Dynamic optimization solutions belong to the probabilistic methods and they are different from randomized algorithms. In this paper, the dynamic path planning problem is solved with evolutionary methods, for compensating the existing deficiencies of the well-known dynamic path planning approaches [1,2]. The comparison of the success of the proposed evolutionary solution with the other algorithms is also presented.

Path planning, in general means, is a kind of problem which aims to find the least cost path from an initial point to a target point in a map. There may be different cost criteria like distance, time and security. The criteria that will be chosen differs according to the application field. In this paper, the application field is military transition, so the system finds the results according to the security, velocity, time and distance criteria. Traditional methods used for the standard path planning problem are Dijkstra, Floyd, A-Star and Neural Network. When the constraints of the problem are dynamic and the environment is not stable, these methods are not valid. The effect of the constraints like security and distance may be modeled with the classical optimization techniques, but in a dynamic environment, restarting to find the solution at each graph update increases the operation complexity. Moreover, when new nodes/segments are defined or some of the existing nodes/segments are deleted from the graph, the classical algorithms can not compensate this situation without starting the solution from the beginning. Evolutionary methods consider multi-objectives like distance, security and traffic, due to the objective function. They can also respond to the dynamic environment situations and may offer an appropriate solution approach [3]. One goal of the evolutionary programming techniques is to propose solution to multi-objective, dynamic problems [4]. Evolutionary algorithms reduce the number of operations in dynamic environments, because they don't restart the solution at each update, they trend to protect and extend best-fit individuals according to the changed conditions [5, 6].

2. Evolutionary computation method

Evolutionary programming is a collection of methods for the automatic generation of computer programs that solve carefully specified problems, via the core, but highly abstracted principles of natural selection. Genetic algorithms are implemented as a computer simulation in which a population of abstract representations of candidate solutions to an optimization problem evolves toward better solutions [7]. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated,

978-1-4244-2881-6/08/$25.00

multiple individuals are stochastically selected from the current population, and modified to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

3. Application of the proposed heuristic approach to military transition problem

Route planning and navigation control based on evolutionary programming concepts can be designed as a general, flexible and adaptive technique. By integrating the planning process in evolutionary algorithms, definition of the different optimization criteria, dynamic update of the constraints, domain specific evolutionary operators, and control of the dynamic obstacles may be handled. Evolutionary algorithms, in comparison to the classical optimization methods, are more effective for discontinuous and noisy functions [8].

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 Figure 1.

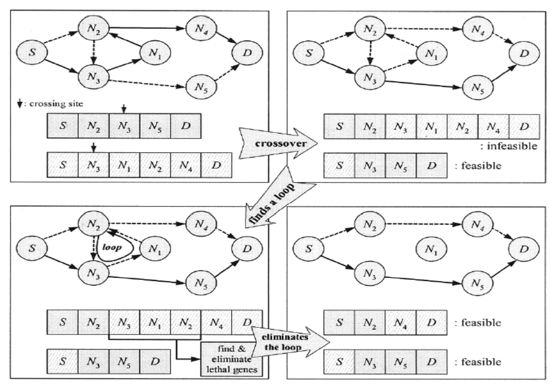

Figure 1. Crossover and repair operators

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 Figure 2.

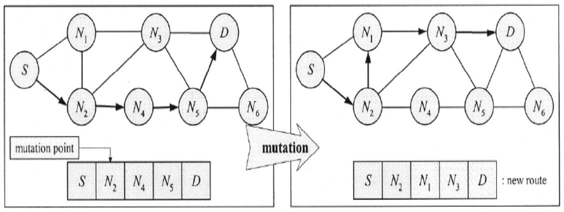

Figure 2. Mutation operator

4. Simulation studies

Simulations are related with the transition of the air vehicles between waypoints. The most secure, shortest and the smoothest transition of the air vehicles from an initial waypoint to a target waypoint is planned. In the city map shown in Figure 3, the links between the cities are represented by vectors of distance, security and altitude. As the flight plan is generated, the altitudes of the waypoints are also considered and 3-D graph solution is planned. The proposed evolutionary method aims to find the shortest, most secure and the smoothest route for the initial constraints. As the problem considers distance, height and security conditions, the problem is multi-objective optimization problem. The initial solution of the dynamic system for the initial conditions is shown in Figure 3.

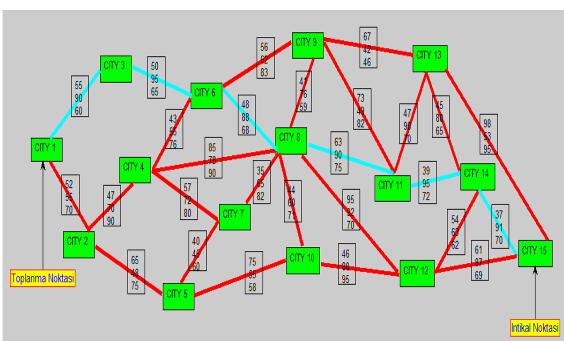

Figure 3. Initial Transition Plan (City 1–City 15)

When the system is in this condition, the links between city 8 – city 9 and city 8 – city 11 are disabled dynamically, the new transition flight plan for the air vehicle is shown in Figure 4.

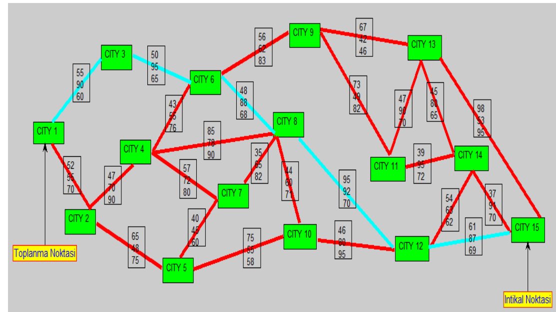

Figure 4. Updated Transition (City 1–City 15)

The system converges to the solution in the dynamic environments without processing from scratch. Analytical approaches repeat the calculations from scratch because an update in the constraints change the matrix used to find the solution. This property is an important characteristic of the proposed algorithm. Evolutionary methods may succeed this by problem specific fitness function and selection operators [11]. The fitness function used in the proposed algorithm is;

Fitness (I) = $\sum$ (1/distance + 1/(100-security) + 1/height_difference) / (number of segments)

The fitness of the individual chromosome is calculated by using the distance, height difference and security values of the segments in the path. Height difference is the difference of altitudes of a segment and the altitude of the previous segment. The height difference is not considered for the first segment of the route. The effects of metrics like security and distance to the solution may be modeled with the classical optimization techniques, but in the dynamic environments like battle scenarios, at each condition change, the algorithm starts the solution from the scratch and increase the operation complexity and calculation time, so they are infeasible [12]. Furthermore, the insertion/deletion of the nodes or segments from the graph changes the matrix of the topology, so they have to start the solution from scratch. The proposed algorithm responds to the concurrent modifications and the best fit individuals according to the modified conditions are spread over the population.

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 update.

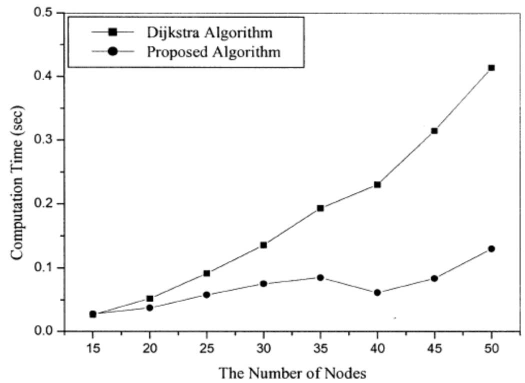

Figure 5. Run-Time Performance

The most recent dynamic path planning solution techniques are Ramalingam and Reps [13], Franciosa et al. [14], and Frigioni et al. [2]. The existing fully dynamic algorithms process unit changes to topology one modification at a time, but when there are several such operations in the environment simultaneously, these algorithms are quite inefficient.

5. Conclusion

Route planning is an optimization problem requires finding a feasible path between an initial and target points without collision. Deterministic algorithms like Floyd, Dijkstra or heuristic methods like neural network and A-Star may be used for the solution of this problem in static schemes. However when dynamic concurrent modifications occur, these algorithms need a longer time to adapt to the new conditions. When nodes or segments in the graph are modified, the problem must be solved from scratch with new constraints. 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.

In this paper, the proposed heuristic approach is tested with the flight plan scenarios. Knowledge based heuristic operators are designed for crossover, mutation, repair operators. The algorithm can be used in both static and dynamic environments. But in static environments, the analytical approaches are preferred. The proposed algorithm provides advanced search speed, quality and flexibility in dynamic schemes.

6. References

[1] Misra S., Oommen B. J., "Dynamic algorithms for the shortest path routing problem: learning automata-based solutions". IEEE Transactions on Systems Man, 2005.

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

[3] Hocaoglu C., Sanderson A. C., "Planning multi-paths using speciation in genetic algorithms", IEEE Int. Conf. Evolutionary Computation, Nagoya, Japan, 1996.

[4] X. Hue, "Genetic algorithms for optimization: Background and applications", Edinburgh Parallel Computing Centre, Univ. Edinburgh, Edinburgh, Scotland, 1997.

[5] Branke J., Salihoglu E., Uyar S., "Towards an Analysis of Dynamic Environments", GECCO 2005: Genetic and Evolutionary Computation Conference, ACM Press.

[6] Elshamli A., Hussein A., Areibi S., "Genetic Algorithm for Dynamic Path Planning", Proc. Canadian Conference on Electrical and Computer Engineering. pp. 677-80, 2004.

[7] Eiben A. E., Smith J. E., Introduction to Evolutionary Computing, Springer-Verlag, Berlin Heidelberg New York, 2003.

[8] Back T., Fogel D. B., Michalewicz Z., "Handbook of Evolutionary Computation", Oxford Univ. Press, London, U.K, 1997.

[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] Uyar A. S., Harmanci A. E., "Preserving Diversity Through Diploidy and Meiosis for Improved Genetic Algorithm Performance in Dynamic Environments", Lecture Notes in Computer Science, Vol. 2457, pp.314-323, 2002.

[12] Misra S., Oommen B. J., "An efficient dynamic algorithm for maintaining allpairs shortest paths in stochastic networks", IEEE Transactions on Computers, 2006.

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

[14] 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.