JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 6, April 2009

Path Planning of Unmanned Aerial Vehicles using B-Splines and Particle Swarm Optimization

Jung Leng Foo* , Jared Knutzon† , Vijay Kalivarapu‡ , James Oliver§ , and Eliot Winer¶
Iowa State University, Ames, IA 50011

DOI: 10.2514/1.36917

Military operations are turning to more complex and advanced automation technologies for minimum risk and maximum efficiency. A critical piece to this strategy is unmanned aerial vehicles. Unmanned aerial vehicles require the intelligence to safely maneuver along a path to an intended target and avoiding obstacles such as other aircrafts or enemy threats. This paper presents a unique three-dimensional path planning problem formulation and solution approach using particle swarm optimization. The problem formulation was designed with three objectives: 1) minimize risk owing to enemy threats, 2) minimize fuel consumption incurred by deviating from the original path, and 3) fly over defined reconnaissance targets. The initial design point is defined as the original path of the unmanned aerial vehicles. Using particle swarm optimization, alternate paths are generated using B-spline curves, optimized based on the three defined objectives. The resulting paths can be optimized with a preference toward maximum safety, minimum fuel consumption, or target reconnaissance. This method has been implemented in a virtual environment where the generated alternate paths can be visualized interactively to better facilitate the decision-making process. The problem formulation and solution implementation is described along with the results from several simulated scenarios demonstrating the effectiveness of the method.

Nomenclature

C	total cost function for a path
C_T, C_L, C_R	threat, fuel, and reconnaissance components cost for a path
c_1, c_2	first and second confidence parameters for PSO
K_T, K_L, K_R	weighting factors for threat, fuel, and reconnaissance components cost

Received 30 January 2008; revision received 8 December 2008; accepted for publication 19 December 2008.

* Post-Doctoral Research Associate, Department of Mechanical Engineering & Human Computer Interaction, Virtual Reality Applications Center, 2274 Howe Hall, Iowa State University, Ames, IA 50010, USA, foo@iastate.edu, AIAA Member.
† Lockheed Martin Aeronautics Company, Palmdale, CA, USA, jared.knutzon@gmail.com.
‡ Post-Doctoral Research Associate, Department of Mechanical Engineering & Human Computer Interaction, Virtual Reality Applications Center, 2274 Howe Hall, Iowa State University, Ames, IA 50010, USA, vkk2@iastate.edu, AIAA Member.
§ Professor, Department of Mechanical Engineering & Human Computer Interaction, Virtual Reality Applications Center, 2274 Howe Hall, Iowa State University, Ames, IA 50010, USA, oliver@iastate.edu.
¶ Assistant Professor, Department of Mechanical Engineering & Human Computer Interaction, Virtual Reality Applications Center, 2274 Howe Hall, Iowa State University, Ames, IA 50010, USA, ewiner@iastate.edu, AIAA Member.

L	length of path
M	number of control points for B-spline curve
N	number of line segments that define the B-spline curve
$\boldsymbol{N}(\boldsymbol{u})$	bernstein basis function for B-spline curve
$\boldsymbol{pu}$	parametric equation for B-spline curve
$\boldsymbol{u}$	set of line segments for B-spline curve
$\boldsymbol{V}$	velocity vector for particle swarm optimization (PSO)
w	inertia weight for particle swarm optimization
$\boldsymbol{X}_i$	ith design variable in an optimization objective function in PSO
$\boldsymbol{x}$	knot vector for B-spline curve
Z_{T}, Z_{R}	threat zone and reconnaissance zone
λ_w	decay factor for inertia weight for PSO

I. Introduction

MILITARY combat of the future will become highly dependent on the use of unmanned aerial vehicles (UAVs). In recent years, there has been rapid development in UAV technology such as swarm communication, command and control, and developing usable interfaces [1]. The complexity in UAV technology is rapidly growing, and according to the Department of Defense (DOD) Roadmap [2], by the year 2012 it is estimated that F-16 size UAVs will be able to perform a complete range of combat and combat support missions. Thus, the ground control station, the human operator's portal to the UAV, must evolve as UAVs grow in autonomy. The ground control station must facilitate the transformation of the human from pilot, to operator, to supervisor, as the level of interaction with UAVs moves to ever-higher levels. As humans interface with UAVs at more abstract levels, a UAV will be trusted to do more [3]. To develop and maintain that trust, a human must be able to understand a UAV's situation quickly. Future ground control stations will need to provide an operator with situational awareness and quality information at a glance.

To address the many research issues involved in the command and control that the DOD roadmap requires, a "Virtual Battlespace" at Iowa State University was created. In this paper, research into the issue of three-dimensional (3D) path planning for UAVs as part of the Virtual Battlespace project is presented. The method described allows a human operator to focus on selecting an appropriate path from a set of alternate paths produced by the path planner, easing the decision-making process. Using a particle swarm optimization (PSO) algorithm, the task of generating alternate paths is formulated into an optimization problem to solve for three objectives: 1) avoid obstacles such as threats (e.g., surface to air missile sites, tanks, and aircraft), 2) maintain a fuel-efficient path to maximize mission range, and 3) remain as close as possible to original way points for reconnaissance purposes. Because there could be multiple solutions that could achieve these objectives, this can be construed as a multi-modal optimization problem. Thus, the use of an evolutionary algorithm such as PSO allows the path planner to search for the global minimums effectively. The simulated scenarios used to evaluate the path planner showed that the path planner was effective in producing alternate paths that satisfy the three objectives.

II. Background

There are three main areas of research to be discussed in this section of the paper to implement the vision of efficient decision making in multi-objective path planning properly. The first section describes the Virtual Battlespace framework, the second section details the PSO algorithm, and the third section describes current research and techniques of path planning for various autonomous vehicles.

A. Virtual Battlespace

Development of the Virtual Battlespace originated in 2000 when a research team at Iowa State University's Virtual Reality Applications Center (VRAC) collaborated with the Air Force Research Lab's Human Effectiveness Directorate and the Iowa National Guard's 133rd Air Control Squadron. The goal of this preliminary version of the Virtual Battlespace was to develop an immersive VR system for distributed mission training. Virtual Battlespace integrates information about tracks, targets, sensors and threats into an interactive virtual reality environment that fuses the available information about the battlespace into a coherent picture that can be viewed from multiple perspectives

and scales [4,5]. Visualizing engagements in this way is useful in a wide variety of contexts including historical mission review, mission planning, pre-briefing, post-briefing, and live observation of mission training scenarios. The Virtual Battlespace can be displayed with a variety of display devices, from a traditional computer monitor, to a completely immersive VR device such as Iowa State University's C6. The C6, shown in Fig. 1, is a six-walled Cave Automatic Virtual Environment (CAVE^{TM}) [6] display device with each wall consisting of a $10' \times 10'$ stereoscopic screen.

The Virtual Battlespace immerses users in a virtual environment that provides them with greater context and situational awareness of the units under their control as well as the overall mission. By integrating radar tracks and UAV video feeds, the virtual world can provide up-to-date access to the latest real time battlefield information. The virtual world is constructed from a mix of a priori information and real-time sensor feeds to act as an organizing context for the operator. The result is a mixed reality system in which real-world video and radar tracks augment a dynamic virtual world. Using real-world data to augment the virtual world is an inversion of the more typical paradigm of augmented and mixed reality where virtual information is used to enhance real-world data and imagery.

B. Particle Swarm Optimization

To facilitate the search for optimal paths, the particle swarm optimization (PSO) technique was used to produce a large number of candidate paths for evaluation. PSO is a heuristic optimization method that is based on the movement of insect swarms introduced in the mid 1990s by Eberhart and Kennedy [7]. PSO is a population based zero-order optimization method that exhibits several evolutionary characteristics similar to genetic algorithms (GA) [8] and simulated annealing (SA) [9]. These are: 1) initialization with a population of random solutions, 2) design space search for an optimum through updating generations of design points, and 3) update based on previous generations [10].

PSO is based on a simplified model of the social behavior exhibited by the swarming behavior of insects, birds, and fish. In this analogy, a swarm member (particle) uses information from its past behavior (best previous location: *pBest*) and the behavior of the rest of the swarm (the overall best particle: *gBest*) to determine suitable food or nesting locations (local and global optimums). The algorithm iteratively updates the search direction of the swarm propagating toward the optimum. Equations (1) and (2) define the mathematical simulation of this behavior.

$$V_{\text{iter}+1} = w_{\text{iter}} \times V_{\text{iter}} + c_1 \times \text{rand}_p() \times (pBest_i[] - X_i[]) + c_2 \times \text{rand}_g() \times (gBest[] - X_i[]) \quad (1)$$

$$X_{\text{iter}+1} = X_{\text{iter}} + V_{\text{iter}+1} \quad (2)$$

$$w_{\text{iter}+1} = w_{\text{iter}} \times \lambda_w \quad (3)$$

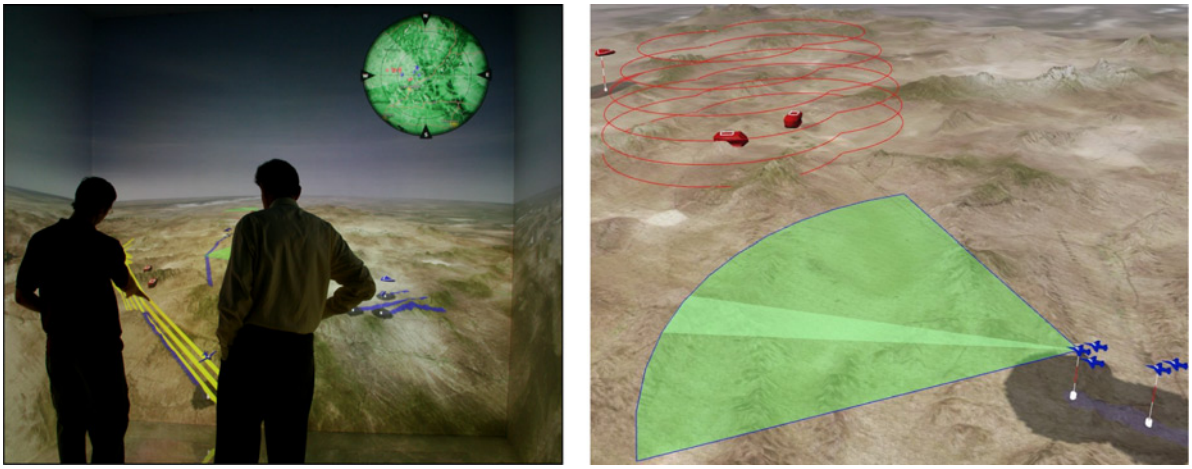


Fig. 1 Virtual Battlespace in the C6 six-wall projection system at Iowa State University's Virtual Reality Applications Center.

Equation (1) represents the velocity vector update of a basic PSO method where $\text{rand}_p()$ and $\text{rand}_g()$ are random numbers generated each for *pBest* and *gBest* between 0 and 1. c_1 and c_2 are user defined confidence parameters. As found by the originators of the method, these values are typically set to 2.0 for effective convergent behavior of the algorithm [11]. '*pBest*' represents the best position of the particle in its history trail, and '*gBest*' represents the best particle location in the entire swarm. w_{iter} is termed 'inertia' weight, and is used to control the impact of a particle's previous velocity on the calculation of the current velocity vector. A large value for w_{iter} facilitates global exploration, which is particularly useful in the initial stages of an optimization. A small value allows for more localized searching, which is useful as the swarm moves toward the neighborhood of the optimum [12,13]. These characteristics are attributed to the swarm by implementing a decay factor, λ_w for the inertia weight, as shown in Eq. (3). Equation (2) denotes the updated swarm location in the design space.

A substantial amount of success has been achieved in utilizing PSO for applications such as aircraft design [14, 15], topology optimization [16], optimal truss structure design [17], wireless network routing problems [18], and optimization in manufacturing and production operations [19,20] to name a few.

C. Path Planning

Path planning is the process of automatically generating alternate paths for an autonomous object based on a set of predefined criteria, when obstacles are detected in the way of the original path. Path planning can be broadly categorized as motion planning and trajectory planning [21]. In motion planning, the focus is getting the object from a starting position to an end position, considering primarily on the translation and rotations needed to get the object there. Recent work has also included the aspects of dynamics, constraints, and optimality. Trajectory planning on the other hand focuses on taking the solution from the motion planning process and then determining a process of moving the object along that path, while still considering the mechanical limitations of that object. Many traditional path planners would implement a brute force search for the next best alternate path, which is an inefficient approach. Later implementations would go on to build a tree structure of possible alternate paths, and then using a branch and bound method to locate the best solution.

There has been extensive research in the area of path planning especially in the artificial intelligence, optimization, and video game communities with most restricted to two-dimensional (2D) paths [22,23]. An example would be the A* algorithm [24], one of the more popular path planning algorithm in the video game/artificial intelligence. The strength of the A* algorithm lies in the ability heuristically to judge or value the best path from point to point. However, with the dynamic nature of a battlefield scenario, the method relies on is constantly changing. This results in the A* method performing inaccurately [25], thus making it ineffective for mission critical path planning purposes. An improvement to the A* algorithm is the D* [26] and the D* Lite algorithm [27]. The D* algorithm is a dynamic version of the A* search algorithm, where it performs multiple searches by modifying previous search results locally to obtain significant search speedup of one to two orders of magnitudes. The D* Lite implements a similar search strategy as the D* but is substantially shorter and simpler to use. Studies performed by the developers showed that even with D* Lite's simplicity, it is at least equally as efficient as the original D*. D* Lite has been successfully implemented in robot path planning situations in unknown terrain [28–30].

Recently, many realized the limitations of 2D UAV path planning and began implementing three-dimensional path planners, taking advantage of the additional two degrees of freedom when generating alternate paths [31–35]. However, the additional information from one extra dimension can create new computational and visualization challenges. An additional degree of freedom results in more alternate path choices, which need to be explored. Another technique of handling multi-dimensional motion planning is the use of probabilistic roadmaps (PRM), introduced in 1996 by Kavraki et al. [36], which overcame the restrictions of higher dimensionality in traditional grid search algorithms. In 1998, Lavalle introduced the concept of rapidly exploring random trees (RRT) [37] that iteratively expanded the tree by adding new samples to the closest existing sample already in the tree.

Improved computation time and more flexible control of the alternate path criteria were observed in instances when optimization has been applied during the path planning process [38–40]. Marti and Qu [41] implemented a path planner based on stochastic uncertainties of the robot's environment and then solving the solution numerically with artificial neural network. The results showed that the solution paths were obtained were close to real time based on the information relayed back by the robot. Another example would be Li et al. [42] with their work on obstacle

avoidance for soccer robots using PSO to find an optimal alternate path based on the robot's current position, shape, and size.

Another form of path planning deals with the control of multiple UAVs by a single user via a cooperative communication scheme between the multiple UAVs [43–46]. During flight, UAVs would exchange information such as altitude and heading, using predefined criteria to control the UAVs collectively as a swarm. Stability issues of UAV swarm formations is still a widely research problem today. Swarm stability ensures that a swarm of UAVs can achieve a group objective as a single cohesive unit without the risk of collision. Methods such as [47,48] i have successfully utilised optimization methods to enhance the efficiency of inter-swarm communications to plan out alternate paths.

Various multi-objective path planners have been developed to accommodate the multiple criteria and dynamic requirements when re-planning a path [49,50]. Methods such as [51,52] have been catered for the transportation of hazardous materials, to optimize the objectives of minimizing risk and at the same time also minimizing the transportation cost. Wu et al. [53] developed a multi-objective path planner that uses a fuzzy logic system in the D* Lite graph search. Their case studies have shown that the method was able to successfully generate alternate paths that satisfy the required criteria.

Another technique used to effectively represent the alternate paths generated by a path planner is using either B-splines. B-splines allow a parametric curved path to be described by only a few control points rather than the entire curve segmented which could be as many as thousands of sections, depending on the length and curvature of the path. Recent methods include modeling the trajectory using splines based on elastic band theory [54,55], and interactive manipulation of control points for spline trajectory generation [56,57]. A description on B-spline is provided in Sect. III-A.

After reviewing various current methods and research being done for path planning of UAVs, it is evident that heuristic optimization methods have been successfully implemented as a means to solve path planning problems. However, there are still issues that must be addressed, namely: 1) developing a path planning process that yields multiple alternate paths quickly and effectively, and 2) allowing a UAV operator (pilot) to visualize the alternate paths and interact with the environment in the decision-making process in an immersive virtual environment. These are the issues that the presented path planner will address in this paper.

The methods reviewed all assumed that the generated alternate path is the only alternative and limits the UAV operator to that one single solution. These situations are useful only if there is a single objective in the path planner such as finding the shortest path or the path with the least obstacles. Also, the methods reviewed did not allow the user to customize the search toward a specific preference such as reconnaissance or fuel efficiency. What if the mission required that the UAV remain close to its original course for reconnaissance purposes? Or, if the UAV operator prefers a safe alternate path while still minimizing to some degree the amount of additional fuel used? The UAV operator needs to be presented with an adequate amount of information and be allowed interactively to make decisions for a more flexible path planning process. To address the aforementioned issues, this paper presents a new method of path planning in 3D space using the PSO algorithm to generate multiple alternate optimal paths quickly based on predefined criteria for visualization in an immersive environment.

III. Method Development

Real-time path alteration is needed when a UAV is presented with an unexpected threat. For example, a UAV could encounter an unexpected surface-to-air missile (SAM) site. When this happens the operator must be alerted to this dangerous situation and be able to quickly re-task the UAV to reduce its threat exposure while considering other factors such as fuel usage. It is important to consider the impact of the immersive environment on this process. In a conventional 2D interface, the application would have to find some way to convey a 3D path in the 2D interface or restrict the path-planning algorithm to a 2D solution; limiting any alternative paths to changes in direction within the same elevation when in reality an aircraft could also change altitude to avoid threats. This limitation is lifted because the Virtual Battlespace operates in an immersive virtual reality environment, which allows true 3D interaction. As such, there is a need for a path planner application that functions in 3D space. With this tool, the operator can focus on the decision to be made as opposed to inferring the true shape of the path.

Without the ability for human intervention of the UAV's path, the Air Controllers and human pilots are solely responsible for maintaining a manageable airspace. Keeping a human in the loop helps prevent catastrophic mistakes by taking advantage of the human's ability to handle and process outside information. The human operators can issue

overall objectives and commands to the vehicles under their control. The issuing of objectives as opposed to exact paths can reduce the amount of awareness needed to control an individual UAV. This reduction could result in more UAVs under the control of a single operator.

The proposed path planner generates alternate paths that are optimized based on three primary objectives, which can be weighted differently based on the operator's preference. The optimization problem is multi-modal in nature, because for every set of obstacles or threats, there can be multiple sets of alternate paths with varying costs. Thus, heuristic evolutionary methods such as PSO or GA would be highly suitable to optimize the generated alternate paths. Literature has showed that while both PSO and GA attained equally comparable high quality solutions, PSO was more computationally efficient on a wide variety of multi-modal unconstrained problems [58]. In addition, PSO has various attractive features such as simplicity in implementation and very few user defined parameters, making it very suitable for this path planning application where the alternate paths need to be generated as quickly as possible. To maintain a human input in the decision-making process, several paths are generated by the path planner and displayed to the operator for inspection and selection. Descriptions of the problem formulation and method development are presented in the following sections.

A. B-Spline Formulation

The generated alternate paths are represented by B-spline curves to minimize computation, because a simple curve can be easily defined by as little as only three control points and this method has been successfully used to model constrained curves [59]. General discussions of B-splines and its variations can be found in most geometric modeling textbooks such as [60]. This section provides a brief description of B-splines to facilitate the path planning optimization problem formulation in the coming sections. A parametric B-spline curve $\underline{\boldsymbol{p}}(u)$, of the order k (of degree $k-1$), is defined by $(n+1)$ control points p_i, knot vector $\boldsymbol{x}$, and by the relationship

$$\boldsymbol{p}(u) = \sum_{i=0}^{n} \boldsymbol{p}_i \mathrm{N}_{i,k}(u) \tag{4}$$

where $N_{i,k}(u)$ are the Bernstein basis functions and are generated recursively using

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } (x_i \leqslant u \leqslant x_{i+1}) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and

$$N_{i,k}(u) = \frac{(u-x)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k-u} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}} \tag{6}$$

The control points define the shape of the curve, and by definition, a low degree B-spline will be closer and more similar to the control polyline (the line formed by connecting the control points in order). The B-splines used in this methodology will be constrained to third-degree B-splines to ensure that the generated curves stay as close to the control points as possible.

B. Proposed Path Planning Algorithm

The path planning process begins with identifying a target location for a specific UAV. Once the current and target positions are defined, this becomes the initial solution of the problem. From this initial solution, a search space is defined to scan and locate other UAVs within range and identify possible threats. The size of the search space is left open to the user's judgment, setting it too large will incur a longer computation time, while having a search space that is too small might cause some UAVs to be unaccounted for. Fig. 2 shows the process flowchart of the path planning using PSO.

C. Alternate Path Objective Function Formulation

Once position data of the UAVs within range are obtained, enemy entities are singled out and a 3D threat zone is generated for each of them. A threat zone is defined as a sphere (a hemisphere for ground vehicles) of radius R_{T}

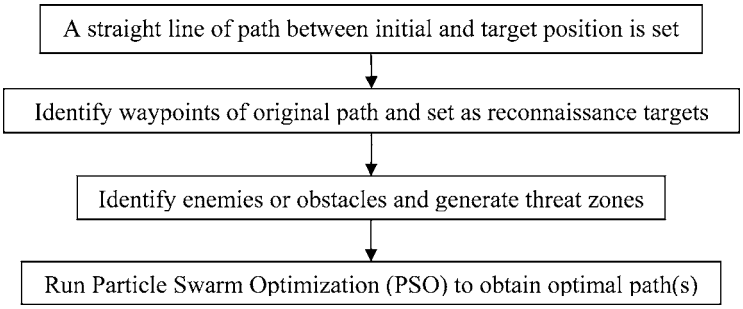


Fig. 2 Flowchart of the path planning process using PSO.

(user defined) surrounding the obstacle that the path needs to avoid. Threat zones are also generated for non-enemy (friendly) entities to avoid collision, but with a smaller radius. In addition, reconnaissance zones are also defined as hemispheres of radius R_R. By default, the value for R_T for an enemy entity is set to be 20,000 feet and for a friendly entity such as other UAVs, the value is set to 5000 feet. The default value for the reconnaissance zone, R_R is set at 2000 feet, but can be changed to suit the UAV operator's preferences.

Formulation of the optimization cost function begins with the description of a B-spline curve to represent the path of the UAV. Consider the B-spline curve $\boldsymbol{p}_0(u_i)$, where u_i is a sequence of line segments forming the curve, that requires re-planning when it violates a threat zone Z_T (in red), shown in Fig. 4. The subscript i denotes the number of line segments that represent the curve. A resulting alternate curve path $\boldsymbol{p}(u_i)$ is generated that avoids the threat zone while still getting as close as possible to be within the reconnaissance zones (in green) and is illustrated by the red curve in Fig. 3.

The shape and curvature of a B-spline curve is defined by a set of control points. To allow for sufficient 'slack' in the curve for a maneuverable alternate path, the number of control points, M, is defined as

$$M = n_T + n_W + 2 \tag{7}$$

where n_T is the number of threats detected, n_W is the number of waypoints for the path, and the additional two control points are used to defined the start and end position of the B-spline curve.

The calculation of the cost function components depends on the number of parametric samples (line segments that form the curve) N that define the resolution of the curve. Here, N is defined as 10 times the value of M and the value of N brings a trade-off effect between accuracy of the curve and computational efficiency. For that reason, the value of N is limited to a maximum of 200 segments for each path. The cost function components are summations of the curve characteristics sampled at the regular parametric intervals

$$u_0 = 0 \quad \text{and} \quad u_i = u_{i-1} + \frac{1}{N-1}, \quad i = 1, 2, \ldots, N-1 \tag{8}$$

where $N = 10 \times M$ and $N < 200$

The initial solution to begin the optimization process is the original path that breaches the threat zone thus violating the constraint, illustrated in Fig. 4 by the blue dashed line. A new path can be computed by running the PSO such that

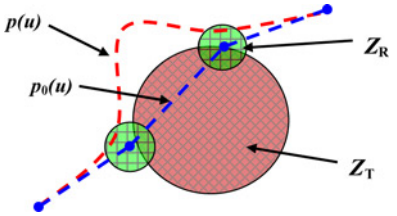


Fig. 3 Two-dimensional illustration of a simple threat zone avoidance problem.

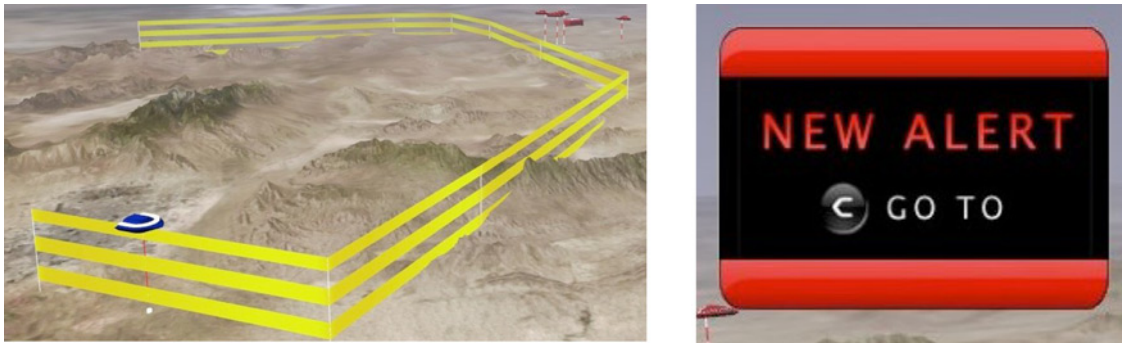


Fig. 4 Illustration of a path in the Virtual Battlespace environment (left) and a threat alert display (right).

the interior control points (between the end-points) satisfy the constraints. To achieve this, the cost function needs to accommodate the preferences of safety, reconnaissance missions, as well as minimal additional fuel usage of the alternate paths. The total cost function is represented by the following components:

$$C = K_{\mathrm{T}}C_{\mathrm{T}} + K_{\mathrm{L}}C_{\mathrm{L}} + K_{\mathrm{R}}C_{\mathrm{R}} \tag{9}$$

where C_{T}: cost owing to proximity of enemy entities and violation of the threat zones

C_{L}: cost incurred from excessive arc length and deviation from the original path

C_{R}: cost incurred by deviating from the reconnaissance locations.

The constants K_{T}, K_{L}, and K_{R}, in Eq. (9), are component weights that determine the relative emphasis of various cost components with respect to the overall cost function. Each weight is normalized between zero and one. If a weight is zero then the corresponding cost function is unimportant for a particular run. All weights sum up to 1.0 in total. These weighted cost components are then added together to form the total cost function of a particular path. If the threat component is of equal importance as the length of the path, both constants will be set equal or have a value of one. Table 1 shows an example of generating a set of three different alternate paths, each with its own preference.

The threat component C_{T} requires a function to determine the distance from a point $\boldsymbol{p}(u_i)$ along the curve $\boldsymbol{p}(u)$ to a UAV inside the threat zone Z_{T} and is denoted here as $d(\boldsymbol{p}, Z_{\mathrm{T}})$. The function will return a positive value if there is a violation of the threat zone, and negative one otherwise. In the case of multiple threat zones, the algorithm will evaluate all threat zones and the largest value of d is used. With this, the threat cost is then defined as

$$d(\boldsymbol{p}(u_i), Z_{\mathrm{T}}) = (\text{Threat zone radius, } R_{\mathrm{T}}) - (\text{Distance between point } \boldsymbol{p}(u_{\mathrm{i}}) \text{ and threat}) \tag{10}$$

$$C_{\mathrm{T}} = \sum_{i=0}^{N-1} C_i, \quad C_i = \begin{cases} |d(\boldsymbol{p}(u_i), Z_{\mathrm{T}})|, & d(\boldsymbol{p}(u_i), Z_{\mathrm{T}}) \geqslant 0 \\ 0 & d(\boldsymbol{p}(u_i), Z_{\mathrm{T}}) < 0 \end{cases} \tag{11}$$

The threat component is calculated based on the assumption that the threat risk increases linearly the closer a UAV gets to the threat. Complex models such as a quadratic model or an exponential model could be used where the threat risk would increase at a faster rate the closer a UAV gets to the threat. The way the objective function for the cost of a path is set up, the threat component will be the dominant component for a threat avoidance path as long as the threat avoidance weight is set to be significantly larger than the other weights, as shown in Table 1. Thus, if a quadratic or exponential model is used for the threat risk assessment, the threat component cost will be higher when

Table 1 Example of component weights used to generate a set of alternate paths

	Threat weight, K_{T}	Fuel weight, K_{L}	Recon weight, K_{R}
Threat avoidance	0.90	0.05	0.05
Fuel efficiency	0.05	0.90	0.05
Reconnaissance	0.05	0.05	0.90

compared with a linear model and will change the path computed. This would not have any influence on the solution strategy, as it would only alter the objective function equation, not the method that solves it.

A significant violation of the threat zone will result in a significant increase in the threat component of the cost function. Because this simple zone violation constraint allows many possible solution curves with unacceptably large length, the second component will simultaneously minimize the curve arc length, providing a solution with the best fit possible along the obstacles.

The curve length component of the cost function is computed using a *chordal* approximation of the total curve length, L, relative to the initial solution obtained from a line connecting the endpoints. The curve length component is expressed as follows

$$C_{\mathrm{L}} = L - L_0, \quad L = \sum_{i=0}^{N-2} |\boldsymbol{p}\,(u_{\mathrm{i+1}}) - \boldsymbol{p}\,(u_i)| \tag{12}$$

L_0 : Length of original path

The curve length component translates as a difference between the generated path and the original path. This represents the additional fuel expense from the alternate path. Should the path planner find a shorter route (regardless if it violates a threat zone), this component will return a negative value, thus turning this component into a reward rather than a cost. The goal is to generate a new path for the UAV that avoids a threat, with the lowest additional fuel expense simultaneously. Currently, the fuel cost component only accounts for increase in distance traveled but does not consider the aerodynamics and propulsion effects of the alternate paths generated, such as restricting fuel usage for a particular generated path. The path planner always generates the best possible path for a given set of objectives and weightings.

A third and final component is for reconnaissance, C_{R}, which is a function to determine the distance from a control point p_i of the curve $\boldsymbol{p}(u)$ to a particular waypoint location Z_{R} and is denoted here as $d(p_i, Z_{\mathrm{R}})$. Because B-spline will curve according to the control points, the control points can be compared with how waypoints are used to define a UAV path. The reconnaissance component increases the objective function when the control point is outside the specified reconnaissance zone. The further the alternate path is from the waypoints, the higher the incurred cost to the objective function. In the case of multiple reconnaissance zones, only the closest reconnaissance zone will be considered for a control point. With this, the reconnaissance cost is defined as:

$$d(p_i, Z_{\mathrm{R}}) = (\text{Distance between control points } p_i \text{ and center of } Z_{\mathrm{R}}) - (\text{Reconnaissance zone radius, } R_{\mathrm{R}}) \tag{13}$$

$$C_{\mathrm{R}} = \sum_{i=0}^{M-1} C_i, \quad C_i = \begin{cases} |d(\boldsymbol{p}_i, Z_{\mathrm{R}})|, & d(\boldsymbol{p}_i, Z_{\mathrm{R}}) \geqslant 0 \\ 0 & d(\boldsymbol{p}_i, Z_{\mathrm{R}}) < 0 \end{cases} \tag{14}$$

In addition to these costs, there are also preset limitations as to how high and how low the UAV can go. Within the optimization algorithm, the altitude of every waypoint is evaluated at every iteration. By default, the altitude cannot be higher than 45,000 feet. The maximum altitude can be changed for different types of UAVs being simulated. Because terrain information has yet to be implemented, the current minimum altitude for the UAV is set to be 2500 feet above sea level, which can be changed to suit different vehicles. If any additional fine tuning is required such as exactly how close a path can come to a threat, bounds may be placed on the design variables.

During the optimization process, the PSO algorithm checks for convergence at every iteration. The optimization process terminates once convergence is reached and the best solution at that iteration becomes the optimal alternate path based on the specific preferences. Convergence is checked by comparing the global best, *gBest* of the PSO algorithm. The optimization problem is considered converged when the difference in solutions was within a tolerance of 0.01 for 10 consecutive iterations.

D. Implementation of PSO Path Planner into the Virtual Battlespace

The purpose of an immersive command and control station is to permit the operator to focus on the overall mission status. As the number of aircrafts under an operator's control increases, it becomes impossible to constantly monitor

Table 2 Color and label representations of generated alternate paths

	Color	Label
Reconnaissance	Blue	A
Threat avoidance	Green	X
Fuel conserving	White	Y

and manage them individually. To facilitate this, an alert subsystem was developed as part of the Virtual Battlespace to alert the operator of any issues. The alert subsystem plays a vital role in reassuring the operator that when UAVs run into situations that require user input, the operator will be made aware of them.

The alert subsystem, seen in the right image of Fig. 4, notifies the operator of the presence of an alert and when the operator chooses to examine an alert posted by a threatened UAV, the operator will see a variety of automatically generated path options. The yellow bands shown in Fig. 4 are altitude indicators for the UAV, where each yellow or transparent band represents 5000 feet above mean sea level (AMSL). In the case of Fig. 4, the UAV is flying at an altitude of just over 25,000 feet AMSL. These path options will appear at a distance corresponding to a default value of 30 s ahead of the UAV's current position and reengage with the path when in a safe region. This lead-time can be adjusted by an operator. These points on the old path are used as the start and end points of the path-planning algorithm. All relevant threats and the start and end points are passed to the path-planning algorithm for it to calculate new candidate paths.

To generate the set of alternate paths, the PSO path planner is initiated to solve for three different sets of values for the component weights. For the purpose of this paper the parameter settings used were those in Table 1. However, an operator can adjust the available weights if additional paths for review are desired. One path is weighted toward minimal fuel expense for an alternate path, the second path makes threat avoidance the preference, and the last path is a middle ground between these options. The operator also has the option to vary these parameters to fit the mission objective. The importance of a UAV's current mission or future mission may demand that the UAV stay on its original path. For this reason the operator is provided with the option to cancel the alternate paths. Additionally, once the UAV is beyond the planned start point the UAV system will assume that it should keep its original path. The generated paths are represented in different colors and labels, as shown in Figs. 6 to 11. For the results from the simulated scenarios presented in this paper, the alternate paths are represented as shown in Table 2.

The paths generated using the path planner currently only accounts for a single UAV and not for swarming UAVs. Cooperation and communication within a swarm of UAVs are not included in the path planner, but when multiple UAVs are present, the path planner could potentially be used multiple times for each UAV and then treating other UAVs as obstacles but with a smaller threat zone, as previously discussed in Sect. III. C.

IV. Results

To test out the designed path planner, several simulated scenarios were created within the Virtual Battlespace. These simulations varied in the number of waypoints and the number of threats, to demonstrate the path planner's efficiency and robustness. The results section will comprise of visual and numerical results from three selected test cases, followed by a performance comparison of the path planner for different scenarios.

A. Test Scenario #1

In the "Pilot" view of the generated alternate paths in the left image of Fig. 5, the paths generated were successful in satisfying its objective. Threat zone hemispheres are marked as a set of red rings. The blue path "A" represents the middle group blend where equal preference is placed on safety and fuel usage. Path "X" (in green) is the path that puts safety as a top priority but still tries to minimize fuel usage as a low priority, and Path "Y" (in white) is the exact opposite which minimizes fuel usage but with a low objective for safety. There is a distinct difference in paths "X" and "Y", as "X" plans a path that is of significantly higher altitude (safer) than "Y" which thrives on flying a shorter deviation from the original path to conserve fuel.

The top-down view in the right image of Fig. 5 illustrates another view of the three alternate paths. When the alternate paths are ready to be inspected by a human operator, a "Choose Path" option prompt appears in the Virtual

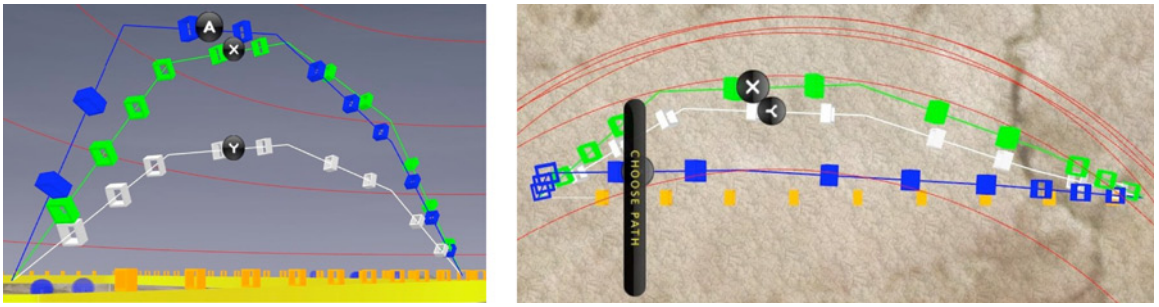


Fig. 5 "Pilot" view of the three resulting alternate paths for a single threat zone (left) and top-down view of the three resulting alternate paths for a single threat zone (right).

Battlespace menu system, and the operator can cycle between three preset views to inspect these paths. The operator also has the freedom to break away from the presets and move around the virtual environment to view these paths. This is an advantage that helps the operator in the decision-making process. A human operator can evaluate these paths from different angles and select an appropriate path for a particular mission.

Once the operator has selected a path, the path is highlighted and a prompt appears in the menu system asking for a confirmation as shown in the left of Fig. 6. This two-step process for path selection is done as a preventive measure in case the operator makes a mistake in selecting the wrong path or decides to go with a different alternate path. When the selected alternate path is confirmed, the path planner replaces part of the original path of the UAV with the alternate path, as shown in the right of Fig. 6.

Table 3 lists the individual component costs for each of the generated alternate paths as well as the original path, for comparison. The reconnaissance and additional fuel costs for the original path are zero because there is no deviation, thus the cost of the original path is made up of only the cost incurred by violating the threat zone.

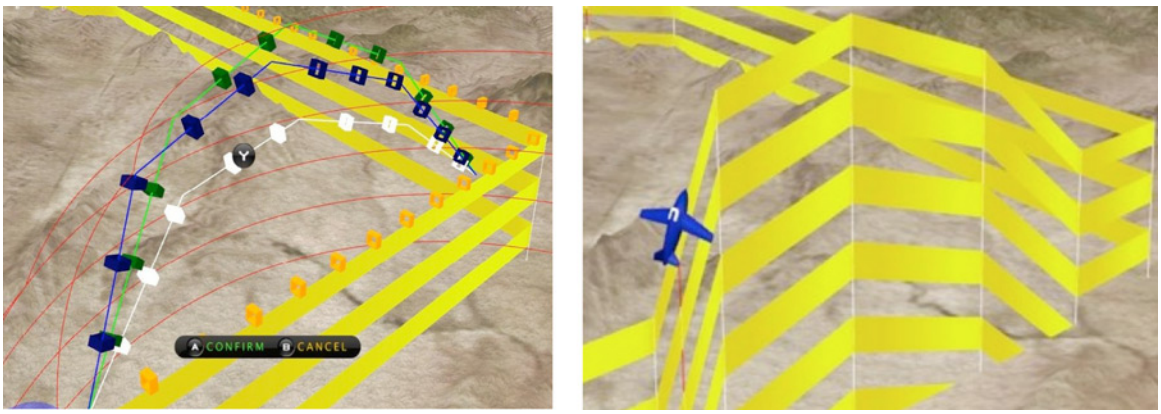


Fig. 6 Highlight of selected path "Y", with the two other alternate paths dimmed. Original path is also shown in yellow (left) and the selected alternate path replaces part of the original path (right).

Table 3 Color and label representations of generated alternate paths

	Recon path	Safety path	Fuel path	Original path
Recon cost (feet)	22,641.31	41,528.80	29,398.44	0.00
Safety cost (feet)	28,755.47	9,246.87	47,901.36	138,227.82
Fuel cost (feet)	8,658.26	10,842.46	6,093.48	0.00
Total cost (feet)	60,055.04	61,618.13	83,393.28	138,227.82

Even with increased fuel component cost and reconnaissance component cost, the total cost of each alternate path is significantly lower than the original path's total cost. This is because the safety component cost increases with every curve segment that violates the threat zone, and depending on how much the path violates the threat zone, this cost can add up significantly if the entire path is deep inside the threat zone. A path that is lower and closer to the SAM site (center of the threat zone) will have a higher safety cost than a path that curves above the SAM site but still within the threat zone. To have a safety cost of zero, the path has to be completely out of the threat zone. When comparing the individual component costs of each path, the reconnaissance path has the lowest reconnaissance component cost, similarly for the safety component cost for the safety path, and the fuel component cost for the minimal fuel usage path, as highlighted in Table 3. This shows that the PSO path planner is indeed effective in determining an alternate path with an optimized solution for a particular preference.

In Fig. 7, the cost of each path during the optimization process is plotted for each iteration. The trend of the plot shows that the PSO algorithm is effective in minimizing the cost of the path and converging to a minimum. The plot showed large reductions in the cost of paths during the early stages of the optimization process when the algorithm is exploring the design space. As the iterations continue and the cost approaches the minimum, the algorithm exploits the particle swarm in the design space to slowly reduce the cost until convergence is met.

B. Test Scenario #2

The second scenario test is one where three threats are situated in close proximity to each other and are in the way of the UAV's original path, as can be seen in the top view of Fig. 8. The threats are represented as red spheres and their corresponding threat zone is represented as a set of red rings. After running the PSO path planner, the resulting alternate paths are shown in Fig. 8. In different views of the generated alternate paths in Fig. 8, these alternate paths were successful in satisfying its objective requirements. The blue path "A" represents the path that thrives to be as close as the original path as possible while still maintaining a small degree of safety and fuel conservation, for reconnaissance missions. Path "X" (in green) is the path that puts safety as a top priority but still tries to minimize additional fuel usage and reconnaissance as a low priority, and Path "Y" (in white) is the third objective which minimizes fuel usage but with low weights for safety and reconnaissance.

The blue path that is weighted for reconnaissance can be seen to be almost identical with the original path except at a higher altitude when approaching the threats. The generated fuel conservation alternate path (in white) has a longer

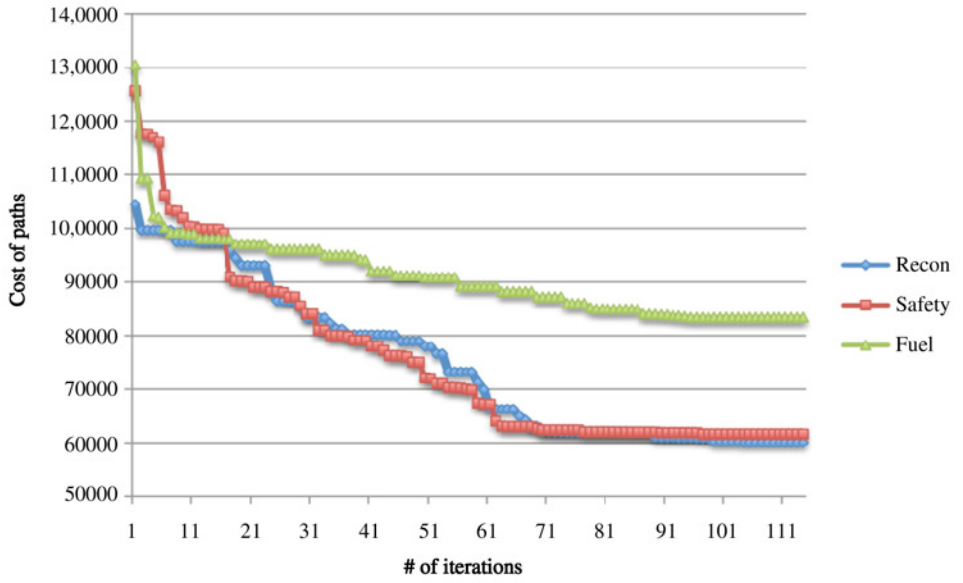
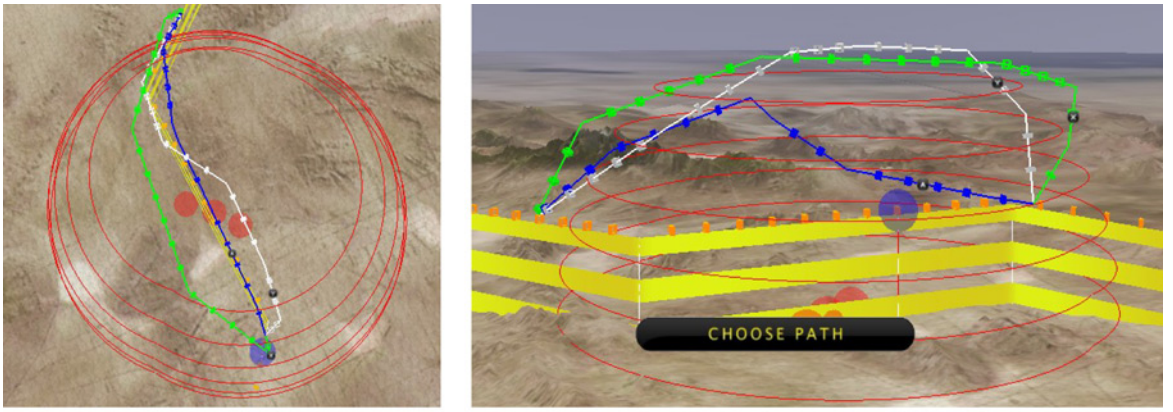


Fig. 7 Plot of the individual cost of the paths during each iteration of the path optimization process using PSO until convergence is met.



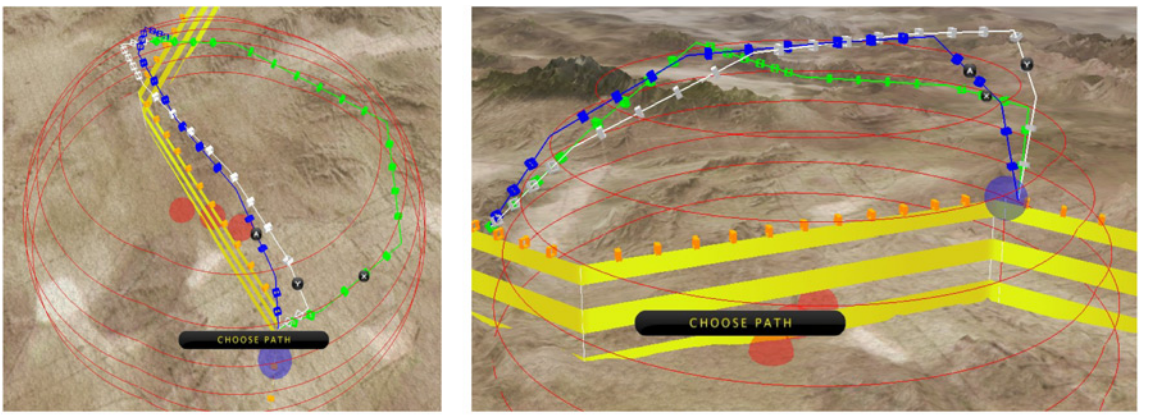
a) Top view of alternate paths. b) Left view of alternate paths.

Fig. 8 First solution of test scenario #2.

path than the reconnaissance path, because the fuel conservation path has equal weights for safety and reconnaissance thus will try to stay outside the detected threat zone. The reconnaissance path, however, will try to stay as close as possible to the original path even though if it means risking violating a threat zone. The green threat avoidance path is distinct from the other two by taking a longer route and staying outside of the detected threat zone.

When the alternate paths are ready to be inspected by a human operator, a "Choose Path" option prompt appears in the Virtual Battlespace menu system, and the operator has the freedom to move around the virtual environment to view these paths. This is an advantage that helps the operator in the decision-making process. A human operator can evaluate these paths from different angles and select an appropriate path for a particular mission.

Because PSO is stochastic in nature, the path planner could produce different solution alternate paths each time for the very same scenario. This is illustrated in Fig. 9, which shows a different set of alternate paths from the same scenario seen in Fig. 8. The design space for this path planning problem is highly multimodal because for a particular objective (reconnaissance, threat avoidance, or fuel conservation), the solution path can be represented in multiple forms, such as having the same length but at different altitudes. Because of the multimodal design space for this path planning optimization problem, the PSO path planner could generate a path that is a local minimum but not the global minimum.



a) Top view of alternate paths. b) Left view of alternate paths.

Fig. 9 Second solution of test scenario #2.

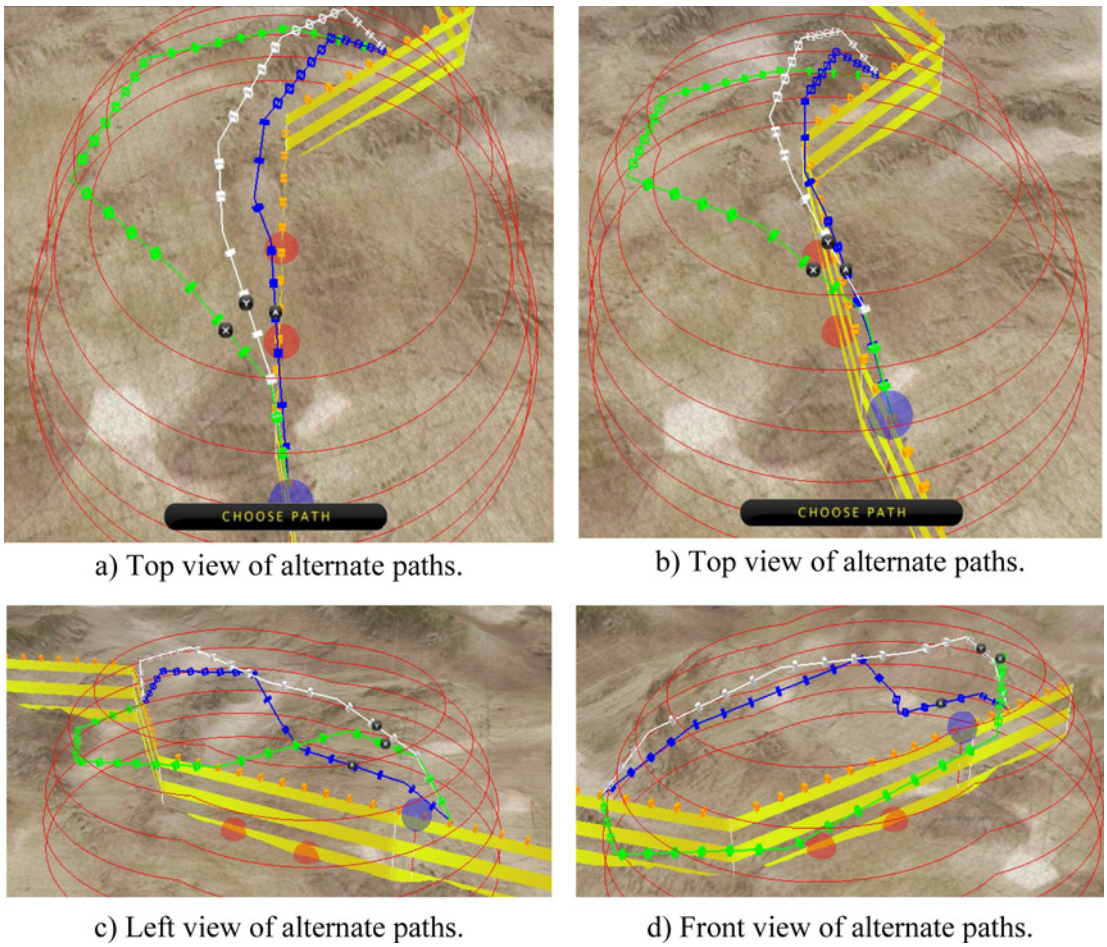
Table 4 Color and label representations of generated alternate paths

	Recon path	Safety path	Fuel path	Original path
Recon cost (feet)	54,764.50	85,387.09	58,444.85	0.00
Safety cost (feet)	21,864.00	7,966.28	23,832.10	176,104.08
Fuel cost (feet)	17,268.45	33,754.63	16,601.80	0.00
Total cost (feet)	93,896.95	127,108.00	98,878.75	176,104.08

As with test scenario #1, the numerical results of test scenario #2 also showed improvements and lower total cost for each alternate path when compared with the total cost of the original path. The same behavior with the individual components is shown in Table 4, where the reconnaissance path has the lowest reconnaissance component cost, similarly for the safety component cost for the safety path, and the fuel component cost for the minimal fuel usage path. Referring to Fig. 9, because the reconnaissance path and the fuel conservation path are very similar, the component costs and the total cost for both paths have very similar values as well.

C. Test Scenario #3

This third scenario has two threats appear in two different locations along the original path, as shown in Fig. 10. The resulting alternate paths generated are also displayed. The alternate paths were generated using the same weights



a) Top view of alternate paths. b) Top view of alternate paths.

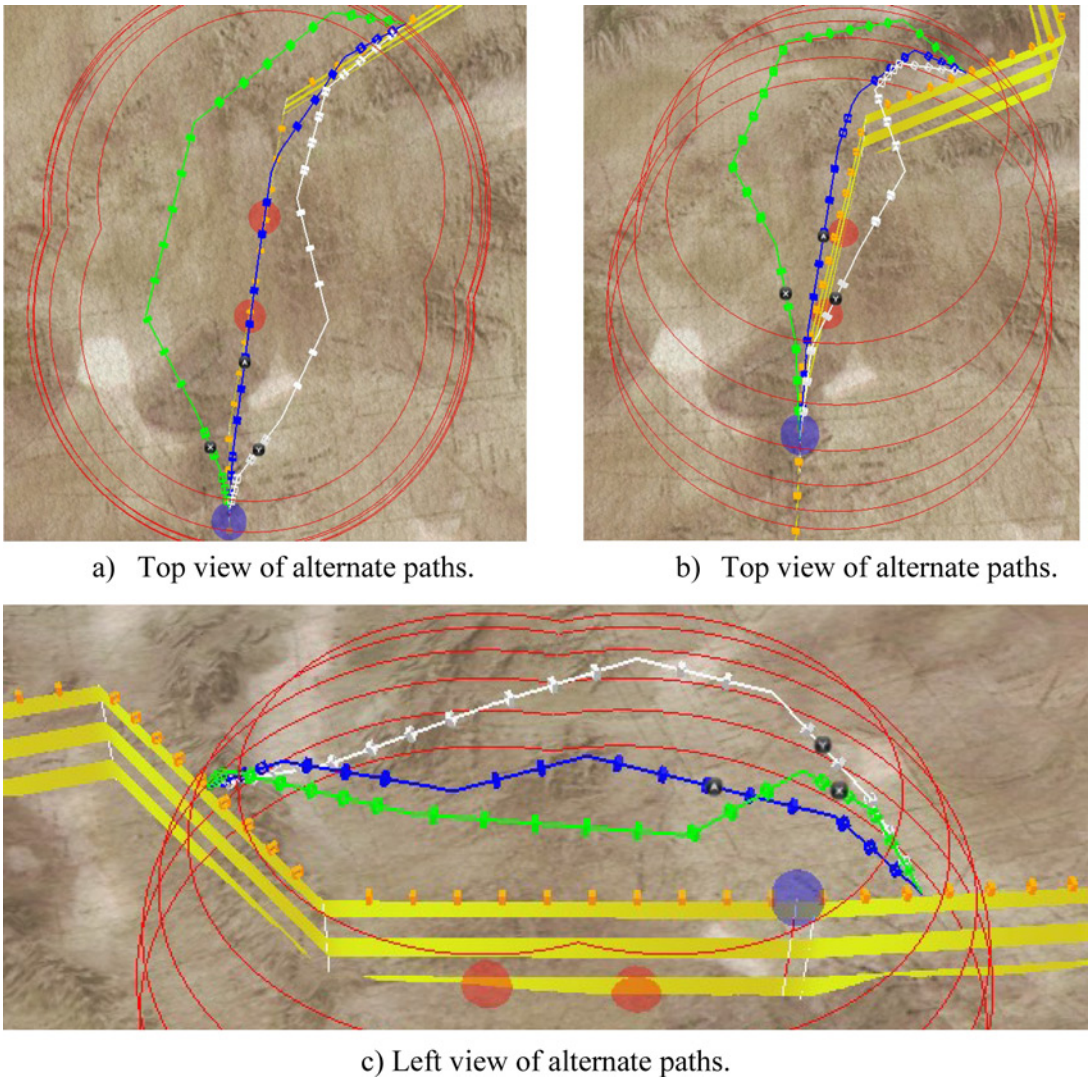
c) Left view of alternate paths. d) Front view of alternate paths.

Fig. 10 First solution of test scenario #3.

as in test scenario #2, shown in Table 1. And similar to the results from test scenario #1, Path "A" (blue) represents the path which places importance on reconnaissance and minimal deviation from the original path, Path "X" (green) is a path that maximizes safety, and Path "Y" is a path that minimizes additional fuel usage of the UAV.

The generated paths for test scenario #3 demonstrate the same behavior to the paths generated for test scenario #2. The fuel conservation path and the reconnaissance path are similar with both having the reconnaissance path having a shorter path than the fuel conservation path and violating the threat zone. As discussed in the results for test scenario #2, this is because the weights in the objective function for the reconnaissance path will force it to stay as close as possible to the original path with minimal deviation, violating the threat zone. The threat avoidance path takes a longer route, wrapping around the threat zone to ensure maximum distance from the threats.

Once again, the stochastic behavior of the path planner is demonstrated when performing a second path planning process on test scenario #3 and a different set of alternate paths were obtained, as shown in Fig. 11. Comparing Figs. 10 and 11, this solution produced a maximum safety path (in green) with a lower altitude but of a further deviation from the original path, thus producing a longer path. In this solution the difference between the reconnaissance path and fuel conservation path is more apparent, as the fuel conservation path in this solution strayed more from the



a) Top view of alternate paths.

b) Top view of alternate paths.

c) Left view of alternate paths.

Fig. 11 Second solution of test scenario #3.

original path compared with the previous solution. Instead of following the original path, the fuel conservation path cuts across diagonally to reach the end point in a shorter distance and avoiding the threats at the same time, as can be seen in the left view in Fig. 11c.

Table 5 lists the individual component costs and the total cost for each of the alternate paths and the original path. As with the previous test scenarios, the numerical results of this test scenario also showed improvements and lower total cost for each alternate path when compared with the total cost of the original path. The PSO path planner was able to successfully generate alternate paths that meet their objectives, as shown in Table 5.

D. Performance Evaluation

This next section evaluates the performance of the path planner in scenarios with varying number of threats and waypoints. In Table 6 and Fig. 12, the average times needed to compute and generate the paths for the test

Table 5 Color and label representations of generated alternate paths

	Recon path	Safety path	Fuel path	Original path
Recon cost (feet)	38,116.00	71,514.35	67,743.92	0.00
Safety cost (feet)	68,653.04	9,102.84	67,099.83	207,419.51
Fuel cost (feet)	19,695.37	41,732.18	17,086.72	0.00
Total cost (feet)	126,464.41	122,349.37	151,930.47	207,419.51

Table 6 Details of each of the test scenarios, listing the number of threats and reconnaissance target waypoints for each scenario, with the average computation time needed to generate alternate paths (in seconds) and the average number of optimization iterations till convergence

Scenario	#1	#2	#3	#4	#5
# of threats	1	2	3	3	4
# of waypoints	2	2	2	3	3
Total objects	3	4	5	6	7
Ave time (seconds)	2.1	3.8	5.2	6.9	8.1
Ave # of iterations	107	138	192	205	223

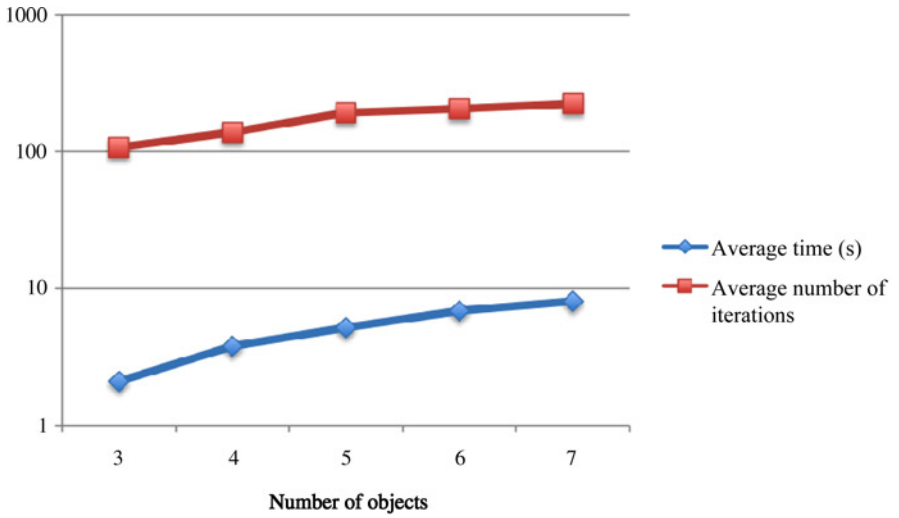


Fig. 12 Computation time for path generation for scenarios with varying number of objects (threats and reconnaissance waypoint targets).

cases are presented. Also included is the average number of iterations taken for the PSO algorithm to converge to a solution.

From Table 6 and Fig. 12, the computation time needed to generate alternate paths is almost linear to the total number of objects (threats and waypoints). This is expected, because the total number of objects translates to the number of design variables for the PSO process, thus increase in objects would mean increase in computation time. The computation times are reasonable, with a complex scenario solved in less than 10 s.

V. Conclusion

A 3D path planner was developed to generate a set of optimized alternate paths to be selected by an operator of a UAV. From the test cases using different simulated scenarios, the 3D PSO path planner successfully generated alternate paths satisfying its respective objectives. The three objectives to either maintain as much of the original path as possible for reconnaissance purposes, to ensure maximum safety, or to maintain maximum fuel conservation were successfully satisfied. These criteria can also be weighted differently to generate paths that are equally biased to all criteria or extremely biased to a specific preference. Most important, these paths were generated in real time to allow for efficient decision making by the UAV operator. By incorporating an immersive environment for path visualization and selection, the decision-making process is more effective and intuitive. The option of selecting a particular path from a set of solutions ensures that a human is still part of the decision-making process. With multiple views to evaluate the generated alternated paths, this allows the operator to make informed decisions based on the current mission objective. The feedback received from experts within the field of UAV control indicate that this is a relevant and promising concept that warrants further investigation.

A. Future Work

The current version of the path planner is limited to only a motion planning implementation, where the paths generated were focused on achieving the UAV from its starting position to its target position while optimizing specific preferences. Planned improvements involve a path planner that is dynamic in nature to incorporate time as a variable when alternate paths are being generated, because the position of threats could change in the future of the alternate path. This would include changing how moving threats are perceived by elongating the threat zone toward the direction that the threat is currently heading, accounting for the probability of encountering the threat in the future. Risk probabilities of the generated alternate paths will also be included in the optimization process.

With the immersive environment of the Virtual Battlespace, the operator can examine the paths with a better sense of spatial awareness. The inclusion of additional information such as the relative costs of the paths for the different objectives can be beneficial to the operator but at the same time can cause information overload and may slow down the decision-making process. Thus, there is current work being done to investigate better ways to represent additional information will still maintaining a level of effective decision making, such as presenting to the operator more than one "best" path for a particular objective and also providing a simplified relative cost for those paths to maintain an efficient decision-making environment.

Also, research is in progress to add physics and aerodynamics to the path planning process to avoid generating paths that might be infeasible for a UAV to maneuver. In addition to that, there is also current research to include terrain information into the problem formulation, where the height of the terrain will be used as constraints during the optimization process. These implementations will bring the path planner closer to applicability in real-world battle scenarios.

Acknowledgement

This research was supported by grants from the Air Force Office of Special Research Labs (AFOSR Award #FA9550-05-1-0384). The authors are grateful for this support.

References

[1] Barry, C. L., and Zimet, E., "*UCAVs Technological, Policy, and Operational Challenges*," Defense Horizons, Center for Technology and National Security Policy, National Defense University, No. 3, October 2001.

[2] US Department of Defense, Unmanned Aerial Vehicles Roadmap 2002-2027, http://www.acq.osd.mil/usd/Roadmap%20Final2.pdf [retrieved July 29, 2008].

[3] Miller, C., "Trust in Adaptive Automation: The Role of Etiquette in Tuning Trust via Analogic and Affective Method," *Proceedings of the 1st International Conference on Augmented Cognition*, ACI, 2005.

[4] Walter, B. E., Knutzon, J. S., Sannier, A. V., and Oliver, J. H., "Virtual UAV Ground Control Station," *Proceedings of AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, AIAA, Reston, VA, 2004.

[5] Ruff, H. A., Calhoun, G. L., Draper, M. H., Fontejon, J. V., and Guilfoos, B. J., "Exploring Automation Issues in Supervisory Control of Multiple UAVs," *Human Performance, Situation Awareness, and Automation: Current Research and Trends*, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, Vol. II, 2004, pp. 218–222.

[6] Cruz-Neira, C., Sandin, D., and DeFanti, T., "Surround-screen Projection Based Virtual Reality: The Design and Implementation of the CAVE," *Proceedings of Computer Graphics (ACM SIGGRAPH) Annual Conference Series*, Addison Wesley, Longman, Reading, MA, 1993, pp. 135–142.

[7] Eberhart, R. C., and Kennedy, J., "A New Optimizer Using Particle Swarm Theory," *6th International Symposium on Micro Machine and Human Science*, IEEE, Piscataway, NJ, 1995, pp. 39–43.

[8] Goldberg, D. E., "*Genetic Algorithms in Search, Optimization and Machine Learning*," 1st ed., Addison Wesley, Longman, Reading, MA, 1989.

[9] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, 1983, pp. 671–680.

[10] Hu, X., Eberhart, R., and Shi Y., "Engineering Optimization with Particle Swarm," *Proceedings of IEEE Swarm Intelligence Symposium*, IEEE, 2003, pp. 53–57.

[11] Engelbrecht, A., *Computational Intelligence, an Introduction*, 2nd ed., Wiley, New York, NY, 2007, pp. 312–315.

[12] Shi, Y., and Eberhart, R., "Parameter Selection in Particle Swarm Optimization," *Proceedings of the 1998 Annual Conference on Evolutionary Computation*, Springer, March 1998.

[13] Shi, Y., and Eberhart, R., "A Modified Particle Swarm Optimizer," *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, IEEE Publications, Piscataway, NJ, 1998, pp. 69–73.

[14] Venter, G., and Sobieszczanski-Sobieski, J., "Multidisciplinary Optimization of a Transport Aircraft Wing using Particle Swarm Optimization," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, 2002.

[15] Pidaparti, R., and Jayanti, S., "Corrosion Fatigue Through Particle Swarm Optimization," *AIAA Journal*, Vol. 41, No. 6, 2003, pp. 1167–1171.

[16] Fourie, P. C., and Groenwold, A. A., "The Particle Swarm Algorithm in Topology Optimization," *Proceedings of the 4th World Congress of Structural and Multidisciplinary Optimization*, Springer, 2001.

[17] Schutte, J., and Groenwold, A., "Optimal Sizing Design of Truss Structures Using the Particle Swarm Optimization Algorithm," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference*, AIAA, Reston, VA, 2002, AIAA Paper 2002-5639.

[18] Yang, S., Huang, R., and Shi, H., "Mobile Agent Routing Based on a Two-Stage Optimization Model and a Hybrid Evolutionary Algorithm in Wireless Sensor Networks," *Lecture Notes in Computer Science*, Vol. 4222, 2006, pp. 938–947.

[19] Onwubolu, G., and Clerc, M., "Optimal Path for Automated Drilling Operations by a New Heuristic Approach using Particle Swarm Optimization," *International Journal of Production Research*, Vol. 42, No. 3, 2004, pp. 473–491.

[20] Rameshkumar, K., Suresh, R., and Mohanasundaram, K., "Discrete Particle Swarm Optimization (DPSO) Algorithm for Permutation Flowshop Scheduling to Minimize Makespan," *Lecture Notes in Computer Science*, Vol. 3612, 2005, pp. 572–581.

[21] LaValle, S. M., "Chapter 1.1: Planning to Plan," *Planning Algorithms*, Cambridge University Press, Cambridge, UK, 2006.

[22] Davis, L., "Warp Speed: Path Planning for *Star Trek:* Armada," *AAAI Spring Symposium*, AAAI Press, Menlo Park, CA, 2000.

[23] Frazzoli, E., Dahleh, M., and Feron, E., "Real-time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 1, 2002, pp. 116–129.

[24] Dechter, R., and Judea P., "Generalized Best-first Search Strategies and the Optimality of A*," *Journal of the ACM*, Vol. 32, No. 3, 1985, pp. 505–536.

[25] Stout, B., "Smart Moves: Intelligent Path Finding," *Game Developer Magazine*, July 1997, pp. 28–35.

[26] Stentz, A., "The Focussed D* Algorithm for Real-Time Re-planning," *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995, pp. 1652–1659.

[27] Koenig, S., and Likhachev, M., "D* Lite," *Proceedings of the National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, CA, 2002, pp. 476–483.

[28] Koenig, S., Tovey, C., and Smirnov, Y., "Performance Bounds for Planning in Unknown Terrain," *Artificial Intelligence*, Vol. 147, 2003, pp. 253–279.

[29] Koenig, S., and Likhachev, M., "Fast Replanning for Navigation in Unknown Terrain," *IEEE Transactions on Robotics*, Vol. 21, No. 3, 2005, pp. 354–363.

[30] Ranganathan, A., and Koenig, S., "A Reactive Robot Architecture with Planning on Demand," *Intelligent Robots and Systems*, Vol. 2, 2003, pp. 1462–1468.

[31] Ryan, A., Zennaro, M., Howell, A., Sengupta, R., and Hendrick, J. K., "An Overview of Emerging Results in Cooperative UAV Control," *Proceedings of the 43rd IEEE Conference on Decision and Control*, IEEE Publications, Piscataway, NJ, 2004, pp. 602–607.

[32] Scherer, S., Singh, S., Chamberlain, L., and Saripalli, S., "Flying Fast and Low among Obstacles," *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2007)*, IEEE Publications, Piscataway, NJ, 2007, pp. 2023–2029.

[33] Langelaan, J., and Rock, S., "Towards Autonomous UAV Flight in Forests," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Reston, VA, 2005.

[34] Sinopoli, B., Micheli, M., Donato, G., and Koo, T. J., "Vision Based Navigation for an Unmanned Aerial Vehicle," *Proceedings of 2001 IEEE International. Conference on Robotics and Automation*, IEEE Publications, Piscataway, NJ, 2001, pp. 1757–1764.

[35] Sasiadek, J. Z., and Duleba, I., "3D Local Trajectory Planner for UAV," *Journal of Intelligent and Robotic Systems*, Vol. 29, 2000, pp. 191–210.

[36] Kavraki, L. E., Svetska, P., Latombe, J. C., and Overmars, M. H., "Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, 1996, pp. 566–580.

[37] LaValle, S. M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning," TR 98-11, Computer Science Dept., Iowa State University, 1998.

[38] Qin, Y. Q., Sun, D. B., Li, N., and Cen, Y. G., "Path Planning for Mobile Robot using the Particle Swarm Optimization with Mutation Operator," *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, 2004, pp. 2473–2478.

[39] Williams, P., "Real Time Computation of Optimal Three-Dimensional Aircraft Trajectories including Terrain-Following," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, 2006, AIAA Paper 2006-6603.

[40] Li, Y., and Chen, X., "Mobile Robot Navigation Using Particle Swarm Optimization and Adaptive NN," *Advances in Natural Computation*, Vol. 3612, 2005, pp. 628–631.

[41] Marti, K., and Qu, S., "Path Planning for Robots by Stochastic Optimization Methods," *Journal of Intelligent and Robotic Systems*, Vol. 22, 1998, pp. 117–127.

[42] Li, W., Liu, Y., and Deng, H., "Obstable-avoidance Path Planning for Soccer Robots Using Particle Swarm Optimization," *Proceedings of 2006 IEEE International Conference of Robotics and Biometrics*, IEEE Publications, Piscataway, NJ,. 2006, pp. 1233–1238.

[43] McLain, T. W., Chandler, P. R., Rasmussen, S., and Pachter, M., "Cooperative Control of UAV Rendezvous," *Proceedings of the American Control Conference*, 2001, pp. 2309–2314.

[44] Bellingham, J. S., Tillerson, M., Alighanbari, M., and How, J. P., "Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environments," *Proceedings of the 41st IEEE Conference on Decision and Control*, IEEE Publications, Piscataway, NJ, 2002, pp. 2816–2822.

[45] Salomon, B., Garber, M., Lin, M. C., and Manocha, D., "Interactive Navigation in Complex Environments Using Path Planning," *Proceedings of the 2003 Symposium on Interactive 3D Graphics*, 2003, pp. 41–50.

[46] Lian, F. L., and Murray, R., "Real Time Trajectory Generation for the Cooperative Path Planning of Multi-Vehicle Systems," *Proceedings of the 41st IEEE Conference on Decision and Control*, IEEE Publications, Piscataway, NJ, 2004, pp. 3766–3769.

[47] Parunak, H. V., Brueckner, S., and Sauter, J., "Digital Pheromone Mechanism for Coordination of Unmanned Vehicles," *Proceedings of 1st International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2002, pp. 449–450.

[48] Sauter, J. A., Matthews, R., Parunak, H. V., and Brueckner, S. A., "Performance of Digital Pheromones for Swarming Vehicle Control," *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2005, pp. 903–910.

[49] Fujimura, K., "Path Planning with Multiple Objectives," *IEEE Robotics and Automation Magazine*, Vol. 3, 1996, pp. 33–38.

[50] Stewart, B. S., and I. Chelsea C. White, "Multiobjective A*," *J. ACM*, Vol. 38, 1991, pp. 775–814.

[51] List, G. F., Mirchandani, P. B., Turnquist, M., and Zografos, K. G., "Modeling and Analysis for Hazardous Materials Transportation: Risk Analysis, Routing Scheduling and Facility Location," *Transportation Science*, Vol. 25, 1991, pp. 100–114.

[52] Leonelli, P., Bonvicini, S., and Spadoni, G., "Hazardous Materials Transportation: A Risk-Analysis-Based Routing Methodology," *Journal of Hazardous Materials*, Vol. 71, 2000, pp. 283–300.

[53] Wu, P., Clothier, R. A., Campbell, D. A., and Walker, R. A., "Fuzzy Multi-objective Mission Flight Planning in Unmanned Aerial Systems," *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision Making*, 2007, pp. 2–9.

[54] Hilgert, J., Hirsch, K., Bertram, T., and Hiller, M., "Emergency Path Planning for Autonomous Vehicles using Elastic Band Theory," *Advanced Intelligent Mechatronics, AIM* 2003, Vol. 2, 2003, pp. 1390–1395.

[55] Sattel, T., and Brandt, T., "Ground Vehicle Guidance Along Collision-free Trajectories using Elastic Bands," *American Control Conference*, ACM, 2005.

[56] Hwang, J., Arkin, R. C., and Kwon, D., "Mobile Robots at your Fingertip: Bezier Curve On-line Trajectory Generation for Supervisory Control," *Proceedings of Intelligent Robots and Systems (IROS 2003)*, IEEE, Vol. 2, 2003, pp. 1444–1449.

[57] Aleotti, J., Caselli, S., and Maccherozzi, G., "Trajectory Reconstruction with NURBS Curves for Robot Programming by Demonstration," *Proceedings of Computational Intelligence in Robotics and Automation*, IEEE, 2005.

[58] Hassan, R., Cohanim, B., de Weck, O., and Venter, G., "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," *Proceedings of 46*th *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA, Reston, VA, AIAA Paper 2005-1897.

[59] Malhotra, A., Oliver, J., and Tu, W., "Synthesis of Spatially and Intrinsically Constrained Curves Using Simulated Annealing," *Journal of Mechanical Design*, Vol. 118, March 1996, pp. 53–61.

[60] Piegl, L. A., and Tiller, W., "B-Spline Curves and Surfaces," *The NURBS Book,* 2nd ed., Spring-Verlag, New York, 1997, pp. 81–116.

Christopher Rouff
Associate Editor