We build from the ground up a computer simulation of a search of missing aircraft. We populate a rectangular search domain with a grid of cells of location density and define an enjoyment function where visitors gain points for going on rides and lose points as they stand in line. We propose two QuickPass systems. In the Appointment System, QuickPasses represent an appointment to visit the ride later that day. In the Placeholder System, a QuickPass represents a virtual place in line. We then choose test cases to represent both systems and run the computer simulation. With each set of parameters, we adjust the probability weights that govern visitor behavior to fit a Nash Equilibrium. The Nash equilibrium adapts the behavior of park visitors to a greedy equilibrium that is not optimal for the group, but is representative of human individuals giving weight to decisions based on what is correlated with giving them an immediate benefit. Our results suggest that it is in the parks best interest to allocate a high percentage of the rides to QuickPass. Reserving too few seats on a ride for QuickPass users can result in average visitor enjoyment being lower than if there were no QuickPass system at all. Both the Placeholder System and the variant of the Appointment System with 75

We are able to parse past 50 years of aircraft accident data, extracting the root cause of the accidents and their typical response (glide, free fall etc.). By parsing these data, we are able to construct distributions of probable crash radius with a relatively high confidence. We run our algorithms on three different types of aircrafts, G280 (small), B737-900ER (medium), and Airbus 380 (large).

Contents

1 Problem Restatement

Concerns over the disappearance of flight MH370 have rekindled interests on how to devise an optimal search plan to maximize our chance at finding the debris of an aircraft lost in a vast open body of water. Given the last known state of the aircraft, we wish to construct a probability distribution of the aircraft debris to and then a search plan that can assist the search-and-rescue teams in allocation of their efforts. Since we fear that the plane have been crashed, we assume no signals can be received from the lost plane, and we ignore the possibility that no foul-play or navigation error could be the only cause for lost of contact with the aircraft.

The specific factors that should be taken into account in obtaining such optimal search plan include the variation in the type of crashed airplane and that of the search agents. Furthermore, we must first determine what constitutes the optimality of a search plan. Once the objective function is determined, the planning problem now turns into one of the optimization and we must determine an efficient and robust way to compute the search plan.

2 Acronyms and Terms

- **SAR**. Search-And-Rescue.
- **MTOW**. Maximum Take Off Weight.
- **USCG**. United States Coast Guards.
- **SAROPS**. Search and Rescue Optimal Planning System.
- **IMTS**. Informational Moving Target Search.
- **Location Density**. Informational Moving Target Search.
- **Search Density**. Informational Moving Target Search.

- **Search Effort**. Informational Moving Target Search.
- **(Detector) Range Law**. Informational Moving Target Search.

3 Assumptions and their Justifications

About the Search Domain and the Missing Aircraft

- **The search domain Ω is a 500km by 300km rectangle of unobstructed ocean.** This rectangle would cover the whole uncertainty range based on the last known state of the lost aircraft for an interval of 15 minutes to 1 hour based on INMARSAT's "Log-on Interrogation" old and newly recommended standards in light of the MH370 accident[**citation**].
- **The missing aircraft is assumed to be still in this domain at $t = 0$.** Although escaping the domain at a later time is allowed.
- **The SAR targets remain clustered.** This means that the targets always stay in the same cell on the discretized grid. This assumption is valid as long as the search is initiated close to the incident time and no serve weather condition causes disturbances.
- **There are buoyant indicator of target location at all time.** Since no underwater search is performed, we assume that either our SAR objective (e.g. survivors, life rafts, parts of crashed debris) remain buoyant throughout our search planning, or our sensor can detect signs of the objectives
- **The local trajectory of concern is straight**. In addition to the obvious smoothness arguments, it is always possible to apply a conformal transform on the entire search domain to obtain a solution based on a curved trajectory.
- **No banking maneuver was made from incident to crash.** This is reasonable for that even in the worse case of gliding due to single engine failure, the average time

from initial to of roughly 11 minutes[**Citation or see derivation later**]. And this time is not long enough to cause trajectory deviations across different search cells.

- **Hijacking or on-board navigation system only problems are not the cause of the incident**. This means that we assume the aircraft is crashed and assumes only the dynamics of the ocean and wind in our search domain. Although hijacking incidents account for nearly 20% of all accidents in past 50 years **Citation!**, the search agents in consideration (e.g. low flying aircrafts and surface vessel) are largely useless in finding a rouge cruising plane. Also see Section 1.

- **The missing aircraft can be accurately modeled as either G280, Boeing 737-900ER, or Airbus 380.** These three types of aircraft are well-known representatives of small private/business jets, medium range commercial flights, and large international flights. Cruise speed and other aircraft form factors (e.g. Lift to Drag ratio) are derived based on this assumption.

- **The crash radius of the aircraft is only a function of the cause of the incident and the type of the aircraft.** In addition, we assume that the historical distribution of the cause of the accidents is a reasonable prior for the current incident at hand, and it is it is invariant with respect to the type of the aircraft[1].

- **Aircraft is operating at MTOW**. Assuming they have cargo and passengers

- **Cruise altitude of all types of aircraft is assumed to be the same**. Describe/Justify.

About Debris Drifting

[1]We do not have an aircraft aficionado at hand to sift through and separate the accident records based on size

- **The local drift direction and speed can be accurately modeled as constant within each cell.** Operationally, the resolution of the cells can be adapted to actual drift data.

- **Nothing outside the search domain drifts back into it.** This assumption only makes searching harder so good for us.

About the Search Agents

- **All agents are commanded and controlled by the central planner at each update interval.** No command and control overhead is assumed for the sake of simplicity.

- **Agents arrive at the boundary of the search domain at $t = 0$.** Search agents are assumed to have arrived on the edge of the search domain at $t = 0$.

- **Unlimited bandwidth between search agent communications**. This is necessary from a planning perspective as to ignore the less than pertinent issues with sensor fusion and coordination. Moreover, this factor is more than likely fixed by the hardware.

- **Search agents are assumed to be either helicopters, UAVs, and surface vessels, all equipped with a bi-variate Gaussian/ definite range law detection probability function.** Moreover, all types of agents are assumed to h Although the problem only mentions "search planes," Marine SAR vessels are very commonly used.

- **Search agents use either magnetometer or camera as their main detector.**

- **Search agents all have a lateral range function of a bivariate Gaussian as a function of their type, altitude, and speed.** Detection follows Koopman's Random search formula.

- **The agents all have null probability of false alarm.** According to [2] and [4], this assumption is a standard one employed by the SAR planning industry. It is an reasonable assumption in that usually false alarms for SAR can be resolved quickly and locally (i.e. reexamining the targets are usually an easy task).

- **The search agents have zero knowledge of local ocean/wind drift information.** Although modern planning softwares used by professional SAR entities (e.g. SAROPS by U.S. Coast Guards) all have existing database to accommodate real-time ocean drifting [1], these data are often not precise, not to mention useless on our fictitious geography.

- **Operational range and refueling problems are neglected.** In operation, search agents can request refueling vehicles etc. and the time is not quite relevant.

- **Agents move in either the horizontal or vertical direction.** Although this is clearly not so realistic operationally, we can always approximate the real search trajectories better by increasing grid resolution.

- **Search agent trajectories are known and executed exactly.** This assumption is alliveated by the fact that we do not assume a definite range law.

4 Literature Review

The problem of finding the optimal search strategy of a target in a fixed region like a body of water, known in the literature as the “Search and Screening Problem” had been extensively researched and analyzed by generations of researchers of operational research. First aroused as a naval problem, Koopman [3] established the foundation on how to approach this problem. In his seminal papers, many reasonable assumptions as well as useful probability functions and formulas (e.g. the random-search formula) are devised and still in use today.

Stone analyzed the overall development of this field up to 1983 in [2].

In [4], Kagan and Ben-Gal composed more recent progress on this old problem. The optimal search plan algorithm used later in this paper is derived from their IMTS methods.

5 Criteria for Optimal Solution

In real life, the only success criterion is whether we can find the target and how long it takes us to do so. However, from the standpoint of search plan optimization, what we want is to maximize the likelihood of detection given a fixed search effort.

6 Debris Density and Search Planning

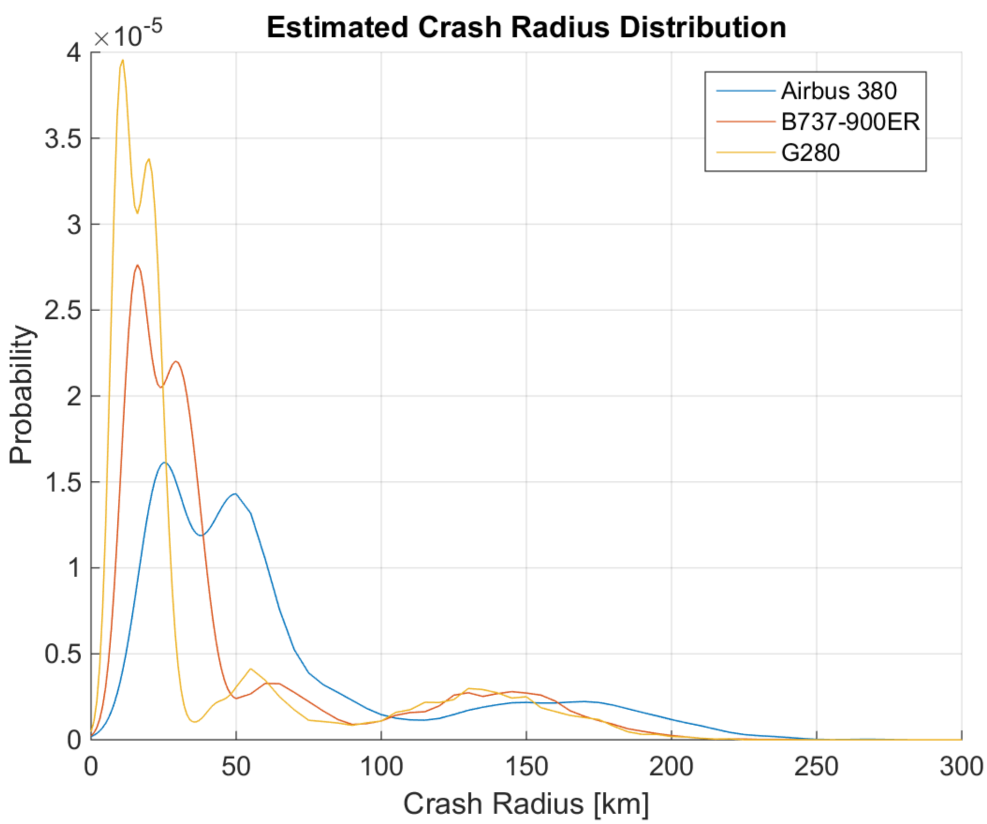

Figure 1: Estimated Crash Radius Distribution of three types of missing aircraft

6.1 Description

6.2 Mathematical Interpretation

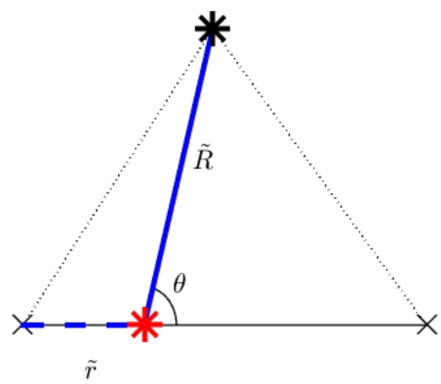

Figure 2: Illustration of the accident geometry. The left $\times$ represents the last known location, the right $\times$ first checkpoint where the aircraft was not found. The incident site is marked by the red $\ast$, while the crash site is marked by the black $\ast$.

The initial location density at each point away from the intended trajectory is calculated as

$$P[crash\,at\,\ast] = \int P[crash\,radius = \tilde{R}(\tilde{r})] \cdot P[Deviation = \theta(\tilde{R}(\tilde{r}))]\,d\tilde{r}$$

Finally, the probability that the aircraft crashed within one cell is obtained through the double integral.

Let us recall the denitions and notation which were applied in the search and screening methods (Section 2.1.1). Assume that the sample space X =x1,x2,...,xn represents a rectangular geographical domain of size (n1,n2), n = n1 n2, and the points xi X, i = 1, 2,...,n, represent the cells in which the target can be located at any time t = 0, 1, 2,.... As indicated above, we index the points of X in a linear order such that for the Cartesian coordinates (j1,j2) of the point xi, its index is given by i = ((j1 1)n1 + j2), where j1 = 1,...,n1 and j2 = 1,...,n2. The target location probabilities at time t, t = 0, 1, 2,..., are denoted by pt(xi) = Prxt = xi, i = 1, 2,...,n, where xt X denotes the target location at time t. As usual, we assume that 0 pt(xi) 1 and n i=1 pt(xi) = 1. The targets movement is governed by a discrete Markov process with transition probability matrix ==ijnn, where ij = Prxt+1 = xj—xt = xi and jn=1 ij = 1 for each i = 1,2,...,n.Ifmatrix is a unit matrix, then the target is static, otherwise we say that it is moving. Given target loca- tion probabilities pt(x), x X, at time t, location probabilities at the next time t + 1are calculated as pt+1(xi) = jn=1 ij pt(xi), i = 1, 2,...,n. The searcher moves over the sample space and at each time t, t = 0, 1, 2,..., looks for a target by testing a neighboring observed area At X of a certain radius r¿0, which is dened by using the metric of the considered geographical domain. Similar to the Singh and Krishnamurthy model (see Section 3.3.1), we do not restrict the size of the available observed areas, but assume that the areas have equal size and the trajectory of the searcher is continuous, as shown in Figure 2.7.

6.3 Comparison to U.S.C.G. SAROPS

In comparison to the direct Monte Carlo / particle filter approach employed by the U.S. Coast Guards' SAROPS system, our search planning algorithm

- **do not use multiple senarios.** due to problem statement; More see 3
- **ss**

6.4 Comparison to SLS Methods

7 Comparison to a Parallel Search Plan

Naively, one can also devise the classic parallel search plan where the search agents travel in parallel lines that each agent's observed area overlaps. Such a plan is analyzed in fair amount of detail by Stone in [2]. In our crude simulation, we assume that the trajectory of the search agents are known and executed exactly

8 Experimental Setup

By changing the type of lost plane, we can notice the difference in prior distribution.

9 Results

10 Sensitivity to Parameters

grid cell size changes cause varying levels of discretization error.

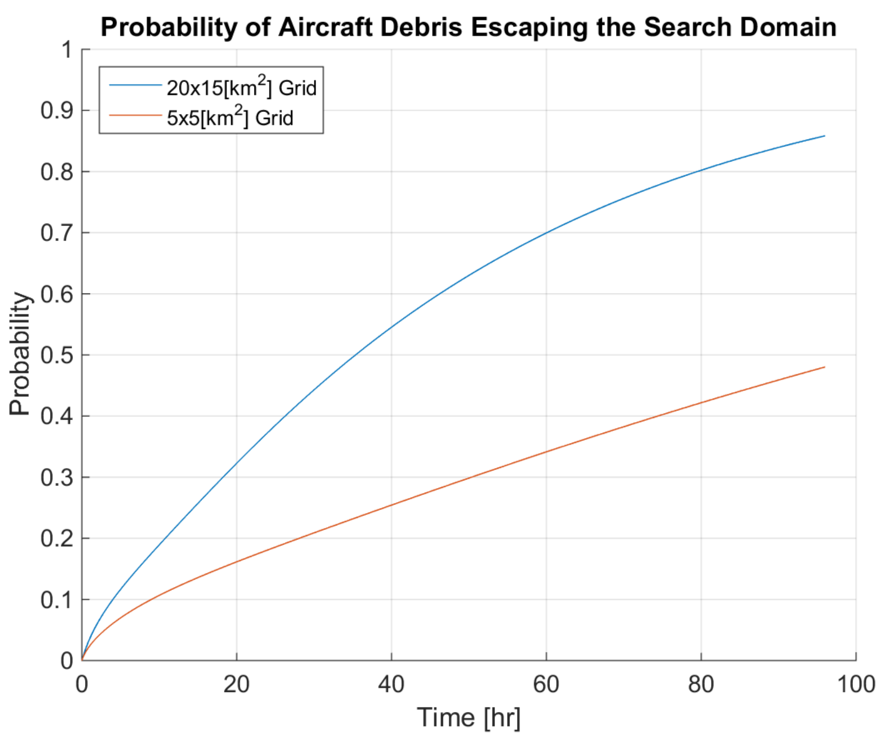

Figure 3: Decrease in grid cell resolution causes higher numerical leakage

In Figure 3, it is evident that reduction in cell resolution causes the Markov drifting simulation leakage.

11 Strengths and Weaknesses

Strengths:

- **simple**. Description.
- **Requires no input data for the ocean drift**. Saves lots of measurement and preparation time.
- **Can be transformed into desired terrain with ease**. Conformal mapping.
- **Short bullet point**. Description.
- **Short bullet point**. Description.

Weaknesses:

- **Require a fine grid resolution for realistic results**. Excessive discretization can cause unwanted error in total escape probability. See Section 10 and Figure 3.
- . Description.
- **very simplified drift modeling**. Description.
- **Short bullet point**. Description.
- **Short bullet point**. Description.

12 Conclusion

- **Recommendation 1**. Why the data says so.
- **Recommendation 2**. Why the data says so.
- **Recommendation 3**. Why the data says so.
- **Recommendation 4**. Why the data says so.
- **Make Sure to form a team of search agents that would not give up the search half way**. One person contributing is $\lll$ two person $\lll$ three person, etc.

References

[1] Kratzke, T.M.; Stone, L.D.; Frost, J.R. "Search and Rescue Optimal Planning System" *Information Fusion (FUSION), 2010 13th Conference on* **DOI:10.1109/ICIF.2010.5712114** (2010) 1-8

[2] Stone, L. D. "The Process of Search Planning: Current Approaches and Continuing Problems." *Operations Research* **31(2)** (1983): 207-233.

[3] Koopman, B.O. "Search and Screening." *Operation Evaluation Research Group Report* **56** (1946) Center for Naval Analysis; See also: "The Theory of Search, I-III" *Operations Research* (1956) **4** 324-246; (1956) **4** 503-531; (1957) **5** 613-626.

[4] Kagan, E.;Ben-Gal, I. "Probabilistic Search for Tracking Targets: Theory and Modern Applications." *WILEY* (2013) 19-313.

[5] idk

[6] U.S. Coast Guard Acquisition Directorate. "Aviation Fact Sheet." `http://www.uscg.mil/acquisITION/programs/pdf/air.pdf`

[7] Damen Shipyards, "NH 1816: a New Type of Search and Rescue boat for KNRM" *Maritime by Holland* (2014) **63** 41-43 `http://products.damen.com/~/media/Products/Images/Clusters%20groups/High%20Speed%20Crafts/Search%20and%20Rescue%20Vessel%201906/Documents/Maritime_by_Holland_2014_Magazine.ashx`

Appendices

Appendix A: Computer Code

main.m :

```
clc; clear; close all;
% constants
gE = 9.81; %m/s2
%% plane specs (B737-900ER; G280; A380)
% Cruise speed (m/s)
B737.Vc = 243;
G280.Vc = 250;
A380.Vc = 262;
% crash distance
% fire, collision, glide, other
A380.R = [24 50 160 80]*1e3; %m
B737.R = [15 30 140 63]*1e3; %m
G280.R = [10 20 135 56]*1e3; %m

%% Assumptions/parameters

% target aircraft make
ACcase = 2;
if ACcase == 1
    AC = B737; acname = 'B737-900ER';
elseif ACcase == 2
    AC = A380; acname = 'Airbus380';
else
    AC = G280; acname = 'G280';
end
% remaining fuel ratio at incident
Fr = .5;
% communication interval/distance
interval = 15*60; %s
rint = AC.Vc*interval; %m
% continuous probability (rtil) riemann sum resolution
Nrtil = 50;
% grid resolution
GRIDcase = 1;
if GRIDcase == 1
    N1grid = 100;
    N2grid = 60;
else
    N1grid = 25;
    N2grid = 20;
end
% # of 1D quadrature points (use even number for symm)
Nquad = 2;
% search domain bounds [-x1 +x1 -x2 +x2]
```

```
bdry = [-150 350 -150 150]*1e3; %m
% reversing probability param
s = .05; q = 1/pi-2*s;

% average debris drift speed
dV = 10; %m/s

%% incident to crash range pdf

% fire, collision, glide, other
Ncrash = [1406, 1901, 901, 498];

% stdev of the statistics (guess)
Rstdev = [.2 .2 .2 .2];

Rhist = [];
for i=1:numel(Ncrash)
    Rhist = [Rhist; randn(Ncrash(i),1)*Rstdev(i)*AC.R(i) + AC.R(i)];
end

% smoothed profile + visuialization
[PR,R]=ksdensity(Rhist,[0:1e3:50e3 55e3:5e3:300e3 310e3:10e3:500e3]);
save([acname '_CrashRadius.mat'],'R','PR');

%% continuous probability at x=(x1,x2)
    rtil = linspace(0,rint,Nrtil);
    Rtil = @(x) sqrt(sum((repmat(x,1,Nrtil)-[rtil;zeros(size(rtil))]).^2));
    theta = @(x) abs(atan2(x(2),x(1)-rtil));
    ptheta = @(x) q + theta(x)*(s-q)/pi;
    pdfx = @(x) interp1(R,PR,Rtil(x)).*ptheta(x);
    %% discritized probability at cell (eu,ev)

    % pre-compute gauss-legendre points and weights
    [u,wu] = gaussquad(Nquad);
    [v,wv] = gaussquad(Nquad);
    Nu = nodefun(u);
    Nv = nodefun(v);

    % total domain area
    Agrid = (bdry(2)-bdry(1))*(bdry(4)-bdry(3)); %m2
    % grid pt construction
    S = Surface(N1grid,N2grid,bdry);
    P = zeros(S.numelements);

    %% loops
    % note the vertical symmetry!!!
    wv = wv*2;
    for ev=1:S.numelements(2)/2
        for eu=1:S.numelements(1)
            for l=1:Nquad/2
                for k=1:Nquad
                    % pullback quadrature pts coordinate
                    [x,y] = S.coords(eu,ev,Nu(:,k),Nv(:,l));
                    Ptemp = sum(pdfx([x;y]))*rint/Nrtil;
```

```
                  P(eu,ev) = P(eu,ev) + wu(k)*wv(l)*Ptemp;
              end
          end
      end
      P(:,S.numelements(2)-ev+1) = P(:,ev);
  end
  %% renormalize and save data
  P = P / sum(P(:));
  save(num2str([ACcase GRIDcase],'prior%d%d.mat'),'P','S');
  % load(num2str([ACcase GRIDcase],'prior%d%d.mat'))

  %% crash probability distribution graph

  Splot = Surface(N1grid,N2grid,bdry/1e3);
  figure(); hold all; grid on;
  plottwoform(Splot,P,3); colorbar;
  xlabel('Tangent Direction [km]'); ylabel('Lateral Direction [km]');
  title('Aircraft Debris Location Density at t=0 hr');
  hold all;
  traj = plot3([-1e6 0 rint 1e6]/1e3, [0 0 0 0], [1 1 1 1],'rx--');
  set(traj,'linewidth',2,'markersize',15)
  saveas(gcf,num2str(GRIDcase,[acname '_PriorDistribution%d.png']));

%% drift/diffusion simulation
  Tsim = 96*3600; %s

  PP = P;
  % update interval that is appropriate
  % i.e. allow only single cell diffusion given grid resolution
  if GRIDcase == 1
      dt = .1*60*60; %s
      Nsim = Tsim/dt; %steps
  else
      dt = .4*60*60; %s
      Nsim = Tsim/dt; %steps
  end
  [Pmove,~] = driftP(S,dt,dV);

  % propogation steps
  tVec = (0:dt:Tsim)/3600; %hr
  % Probability of escape at t
  qt = zeros(Nsim+1,1);

  for t = 1:Nsim
      [PP,qt(t+1)] = next(S,PP,Pmove);
  end
  save(num2str([ACcase GRIDcase],'nosearchEsc%d%d.mat'),'tVec','qt');
  %% Location density if no search initiates
  figure(); hold all; grid on;
  plottwoform(Splot,PP,3); colorbar;
  xlabel('Tangent Direction [km]'); ylabel('Lateral Direction [km]');
  title(num2str(Tsim/3600, 'Aircraft Debris Location Density at t=%d hr'));
  saveas(gcf,num2str(GRIDcase,[acname '_NoSearchDistribution%d.png']));
  %% Graph of escape probability over time
```

```
    figure(); hold all; grid on;
    plot(tVec,qt,'k-');
    xlabel('Time [hr]'); ylabel('Probability');
    title('Probability of Aircraft Debris Escaping the Search Domain');
    saveas(gcf,num2str(GRIDcase,[acname '_NoSearchEscape%d.png']));
%% Search Agent Data

% given 99% detection range find sigma
ncdf = @(sig,Rd) normcdf(Rd,0,sig)-normcdf(-Rd,0,sig);
cdf2sig = @(Rd) fminsearch(@(sig) abs(ncdf(sig,Rd)-.99),Rd/2);

% Marine Vessel (Damen SAR vessel 1816/1906 range= 600km+)
MV.Vs = 15.5; %m/s
MV.Rdetect = 1e3; %m
MV.FA = 0;
MV.sig = cdf2sig(MV.Rdetect); %m

% UAV (Hermes)
UAV.Vs = 49; %m/s
UAV.Rdetect = 5e3; %m
UAV.alt = 5e3; %m
UAV.FA = .05;
UAV.sig = cdf2sig(UAV.Rdetect); %m

% Helicoptor (USCG Dolphin MH65C range= 280km+)
heli.Vs = 82; %m/s
heli.Rdetect = 5e3; %m
heli.alt = 5.5e3; %m
heli.FA = .05;
heli.sig = cdf2sig(heli.Rdetect); %m

%% search agent initial states
% number of agents on one side
Nagent = 100;

% Initial position

% Ps0 = [S.xnodes(S.getglobalboundarynodes) ...
%     S.ynodes(S.getglobalboundarynodes)];

% Detection Probability are assumed to be normal
mvncdf(x1r,x2r,xs,sig);
%% Distributed Search Plan (edge first and chase the highest cell)
    Tsim = 96*3600; %s

    PP = P;
    % update interval that is appropriate
    % i.e. allow only single cell diffusion given grid resolution
    if GRIDcase == 1
        dt = .1*60*60; %s
        Nsim = Tsim/dt; %steps
    else
        dt = .4*60*60; %s
        Nsim = Tsim/dt; %steps
```

```
    end
    [Pmove,Ncell] = driftP(S,dt,dV);

    % propogation steps

    % Probability of escape at t
    qt = zeros(Nsim+1,1);

    for t = 1:Nsim
        [PP,qt(t+1)] = next(S,PP,Pmove);
    end
    %% Location density
    figure(); hold all; grid on;
    plottwoform(Splot,PP,3); colorbar;
    xlabel('Tangent Direction [km]'); ylabel('Lateral Direction [km]');
    title(num2str(Tsim/3600, 'Aircraft Debris Location Density at t=%d hr'));
    saveas(gcf,[acname '_NoSearchDistribution.png']);
    %% Graph of escape probability over time
    tVec = (0:dt:Tsim)/3600; %hr
    figure(); hold all; grid on;
    plot(tVec,qt,'k-');
    xlabel('Time [hr]'); ylabel('Probability');
    title('Probability of Aircraft Debris Escaping the Search Domain');
    saveas(gcf,[acname '_NoSearchEscape.png']);
%% Distributed Search Plan (center first)


%% concentrated "single" agent Search Plan (steepest descent)


%% Monte Carlo evaluation of time of capture


% "particle filter" ?
```

Appendix B: Full-Page Plots