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

Pursuit of a lost plane:

The disappearance of flight MH370 kindled a long debate and serious introspection. The methods used for hunting the plane came into serious questioning when even after months, the searchers were no closer to discovering the plane or its debris. The problem asks to Build a generic mathematical model that could assist "searchers" in planning a useful search for a lost plane feared to have crashed in open water such as the Atlantic, Pacific, Indian, Southern, or Arctic Ocean while flying from Point A to Point B. Like in the case of MH370, we arent receiving any signal from the plane that is feared to have crashed.

The challenge in building a search model is the variation in Plane sizes and the different search techniques used. The problem requires to account for both.

2 Terminologies and Conventions

- **SAR**. Search-And-Rescue.
- **MTOW**. Maximum Take Off Weight.
- $\tilde{r}$. Distance between point of last contact and point of incident.

3 Assumptions and their Justifications

About the Search Domain and the Missing Aircraft

- **The search domain 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**. Although hijacking incidents account for nearly 20% of all accidents in past 50 years **Citation!**, SAR plane (or vessel) detectors are largely useless in finding a cruising rouge plane. Also see problem restatement.

- **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

- **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

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

sensor fusion and coordination. Moreover, this factor is more than likely fixed by the hardware.

- **Types of search agents considered are helicopter, fixed wing UAVs, and surface vessels.** Although the problem only mentions "search planes," it is very common to have sur
- **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 probability of false alarm is assumed to be 0%.** Lack of exact literature references, industry standard assumption [2013 book].
- **The search agents have no knowledge of local drift direction.** Although modern planning softwares used by professional SAR entities (e.g. SAROPS by U.S. Coastal Guards) all have existing database to accommodate real-time ocean drifting, these data are often not precise, not to mention useless on a fictitious geography.
- **Ignore refueling problems.** In operation, search agents can request refueling vehicles etc. and the time is not quite relevant.
- **Agents move either horizontally or vertically.** In operation, search agents can request refueling vehicles etc. and the time is not quite relevant.

4 Literature Review

Stone ... [3]

Koopman ... [2] Random-search formula

Kagan ... [1]

5 Criteria for Optimal Solution

In real life the only success criterion is whether we can find the target and how long it takes to do so.

6 Describe Your Method

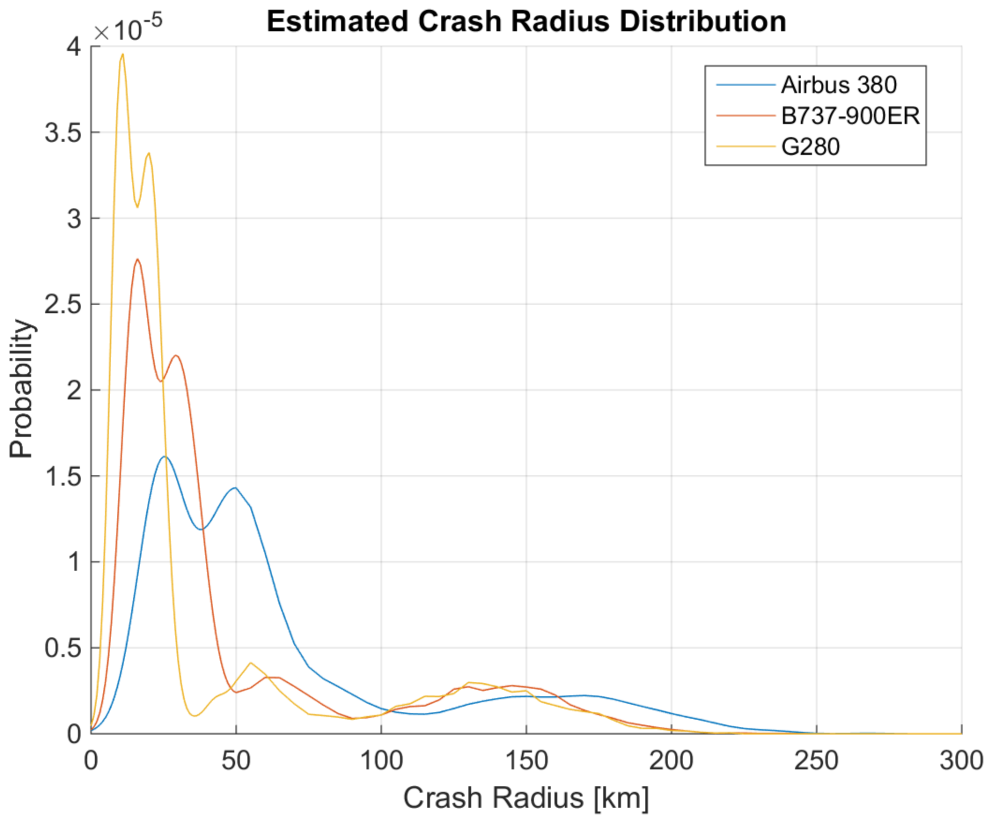

Figure 1: Estimated Crash Radius Distribution of three types of missing aircraft

6.1 Description

6.2 Mathematical Interpretation

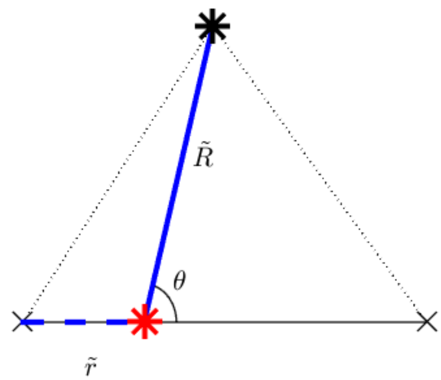

Figure 2: Illustration of the accident geometry. The left $\times$ represents the last known location, the right $\times$ first checkpoint where the aircraft was not found. The incident site is marked by the red $\ast$, while the crash site is marked by the black $\ast$.

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.

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

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

- **do not use multiple senarios.** due to problem statement;

- ss

6.4 Comparison to Newer Methods

Group-testing for multiple targets, but static.

7 Comparison to a "Steepest Decent" Search Plan

also multiple agent/single agent etc.

8 Experimental Setup

9 Results

10 Sensitivity to Parameters

grid cell size changes cause varying levels of discretization error.

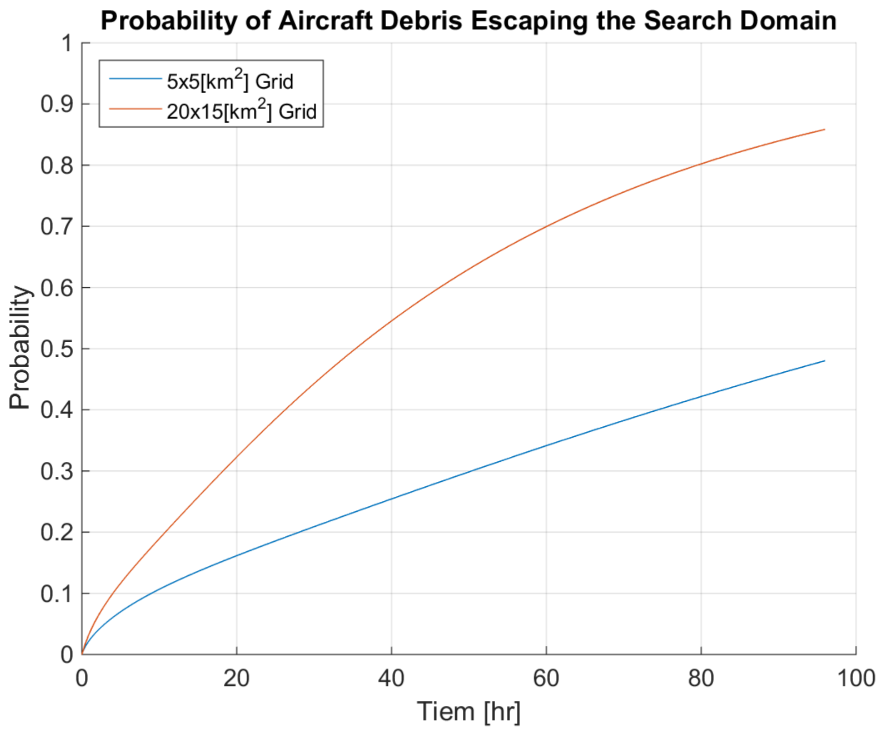

Figure 3: Decrease in grid cell resolution causes higher numerical leakage

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 10 and 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.

References

[1] Kagan, E.;Ben-Gal, I. "Probabilistic Search for Tracking Targets: Theory and Modern Applications." *WILEY* (2013) 19-140.

[2] 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.

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

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 = 1;
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 = 2;
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,'prior%d.mat'),'P','S');
% load(num2str(casenum,'prior%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,[acname '_PriorDistribution.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,[acname '_NoSearchDistribution.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,[acname '_NoSearchEscape.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= 650km+)
heli.Vs = 90; %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