

## Contents

---

- [Communication at Opportunity Sim -- Rectangle Boundary Analysis](#)
- [General Analysis -- Infinite path length, single iteration simulations](#)
- [4, 8, 16, 32 Agent Inf Horizon Test in 20x10 Box](#)
- [4, 8, 16, 32 Agent Inf Horizon Test in 20x40 Box](#)
- [4, 8, 16, 32 Agent Inf Horizon Test in 20x80 Box](#)
- [General Analysis -- 240 path length, multi-start simulations](#)
- [4, 8, 16, 32 Agent Multi-Start Test in 20x10 Box](#)
- [4, 8, 16, 32 Agent Multi-Start Test in 20x40 Box](#)
- [4, 8, 16, 32 Agent Multi-Start Test in 20x80 Box](#)

## Communication at Opportunity Sim -- Rectangle Boundary Analysis

---

By: Patrick Ledzian Date: 10 April 2022 Last update: 28 April 2022

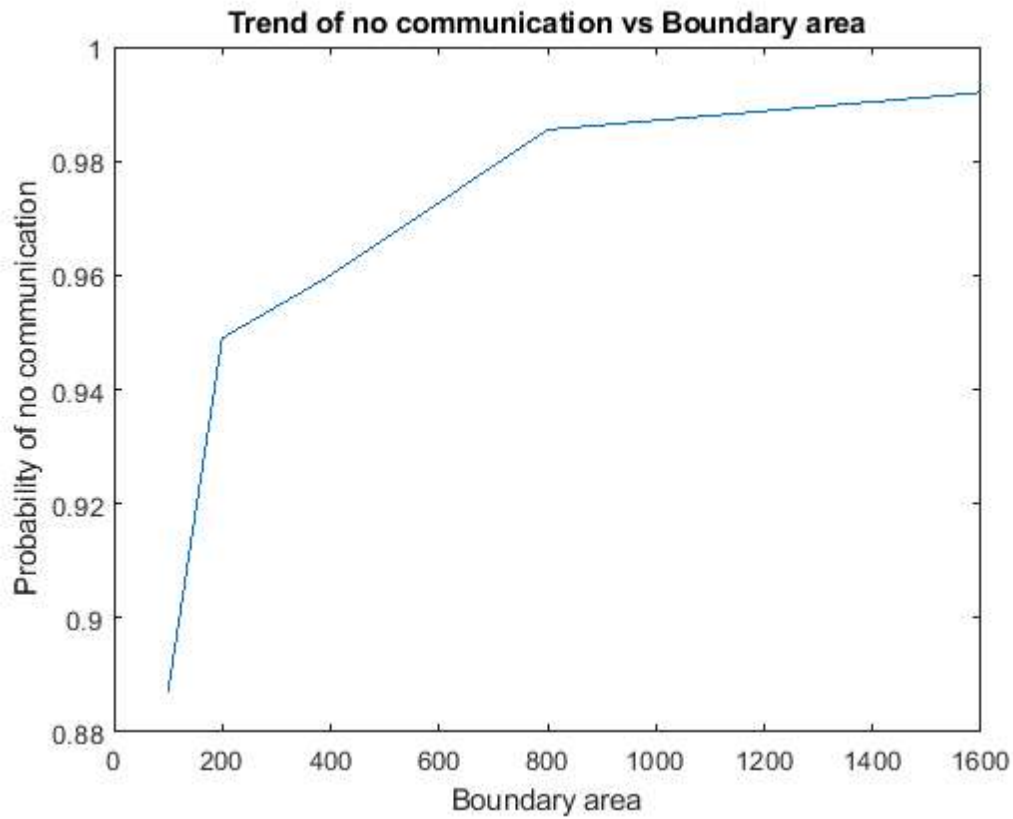
```
% Overview:  
% Generate simulation data to provide insight as to how often agents  
% interact when bouncing around in a closed, axis-aligned rectangle, as  
% well as the relationship between the rectangular skew of the boundary  
% (vs. a perfect square) and the random interactions between agents.  
% Each agent chooses a new random heading (uniform distribution [0, pi])  
% whenever it intersects with a boundary. Longer-term goal is to determine  
% what conditions result in reliably connected communications graphs  
% between agents, independent of agent starting position. As well as to  
% determine how long agents must randomly interact before said  
% communication graphs are connected.  
%  
  
close all  
clear all  
clc  
  
a = SimAnalysis();
```

## General Analysis -- Infinite path length, single iteration simulations

---

With the rectangle axis-aligned boundaries the chance of random comms between non-coordinating agents appears to be stationary with-respect-to the number of agents. However, the transients behavior (rise-time) is shorter the more agents that are present. Also note that as the area of the boundaries increases the chance of communication goes down. A target for future work is to intelligently manipulate the distribution agents choose new random headings from, maybe from local interactions (comms). This should also be balanced with the transient (rise-time) behavior to meet mission needs (vehicle specs, on-scene time, power management, etc)

```
xgen = [100, 200, 400, 800, 1600];  
ygen = [0.887, 0.949, 0.96, 0.9855, 0.992];  
  
figure()  
plot(xgen, ygen)  
xlabel("Boundary area")  
ylabel("Probability of no communication")  
title("Trend of no communication vs Boundary area")  
  
%ylim([0.85 1.0])
```



#### 4, 8, 16, 32 Agent Inf Horizon Test in 20x10 Box

```
% Results:
% 4-Agent case: converges to [0.946, 0.951]
% 8-Agent case: converges to [0.947, 0.953]
% 16-Agent case: converges to [0.946, 0.952]
% 32-Agent case: converges to [0.945, 0.952]
%

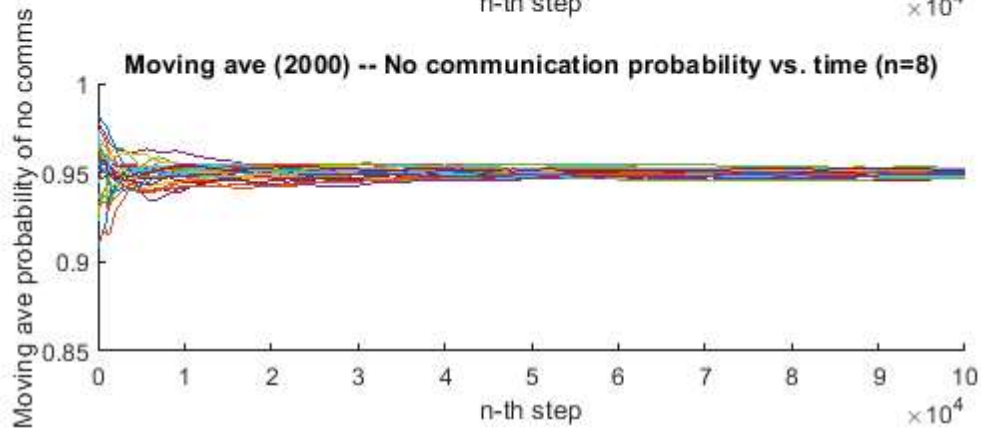
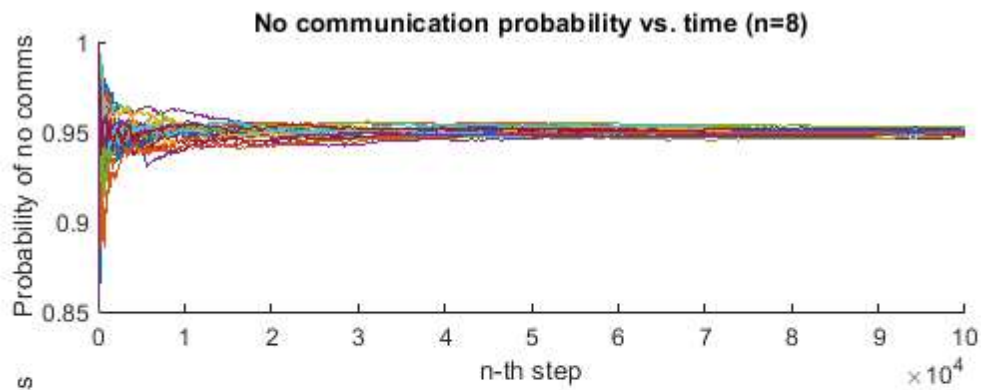
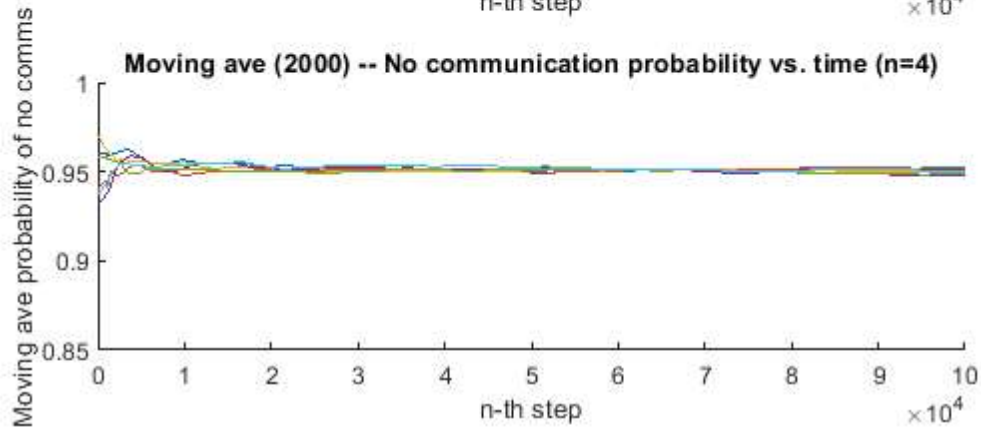
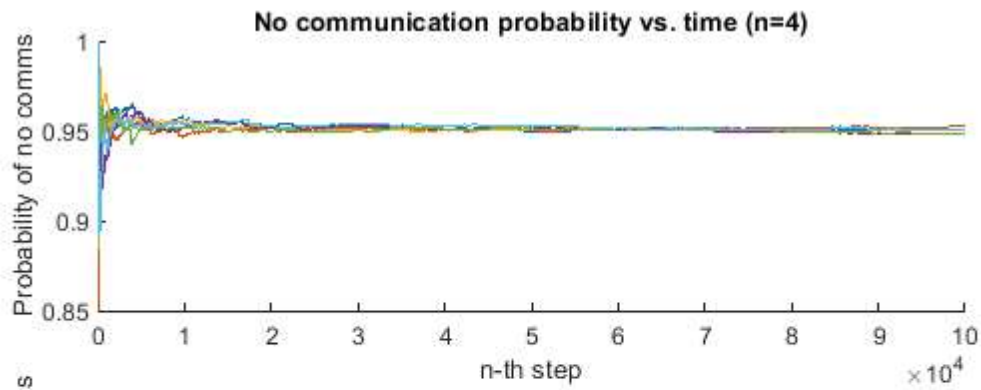
bounds = [0 10 10 0; 0 0 20 20];

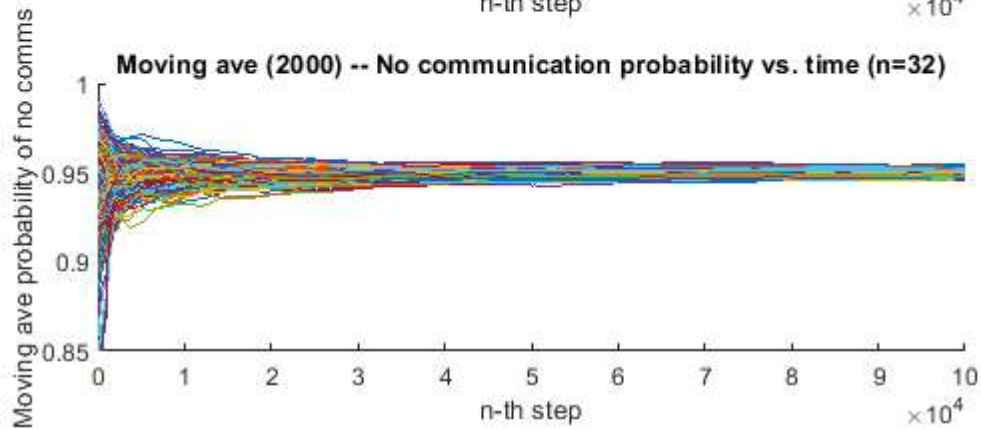
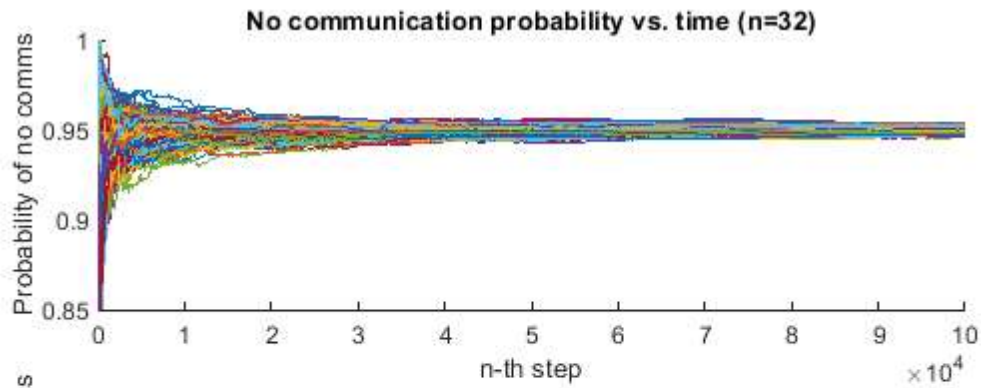
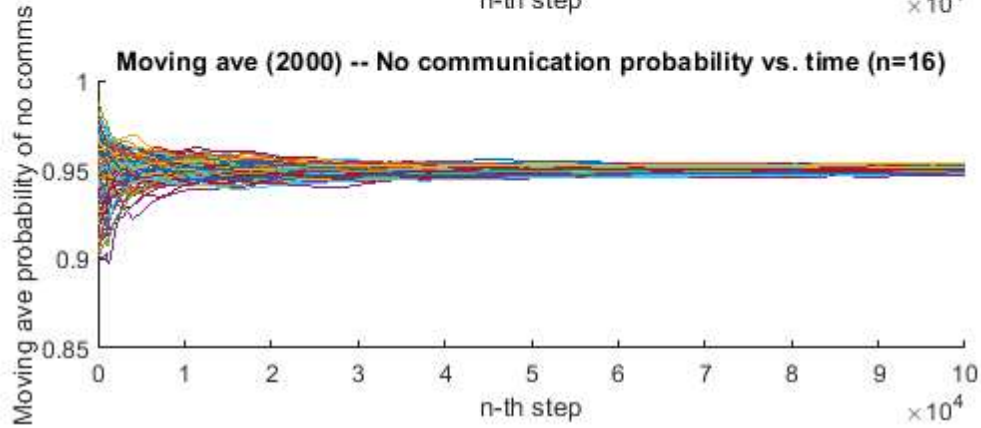
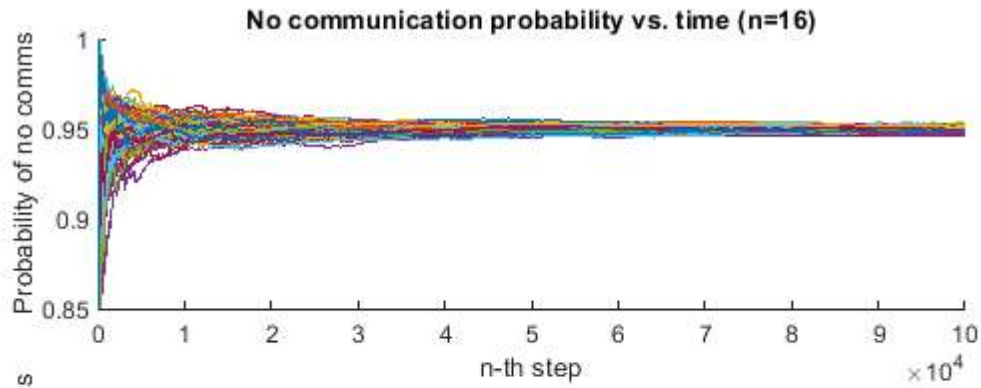
a.steadyStateCommProb(bounds, 4, 100000, 1);
a.steadyStateCommProb(bounds, 8, 100000, 1);
a.steadyStateCommProb(bounds, 16, 100000, 1);
a.steadyStateCommProb(bounds, 32, 100000, 1);

SimAnalysis.clearEmptyFigs();
```

% bounds, numAgents, N, sim\_itr

% closes figures without titles (for empty ones!)





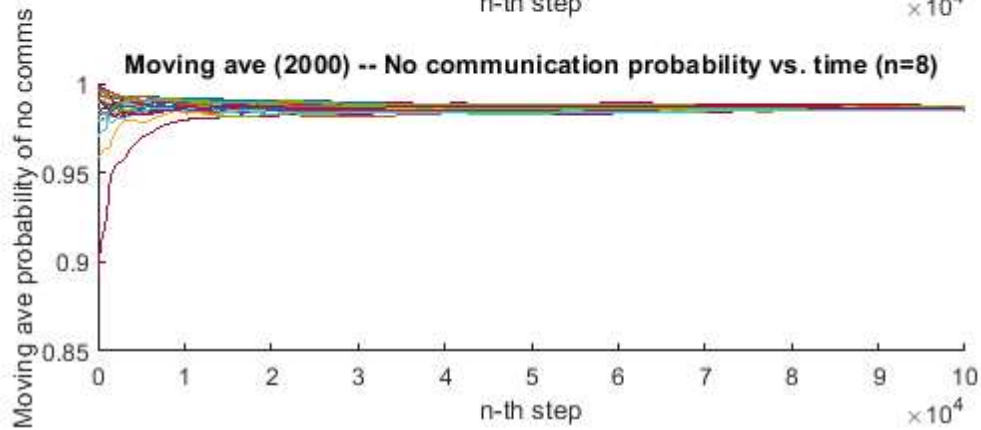
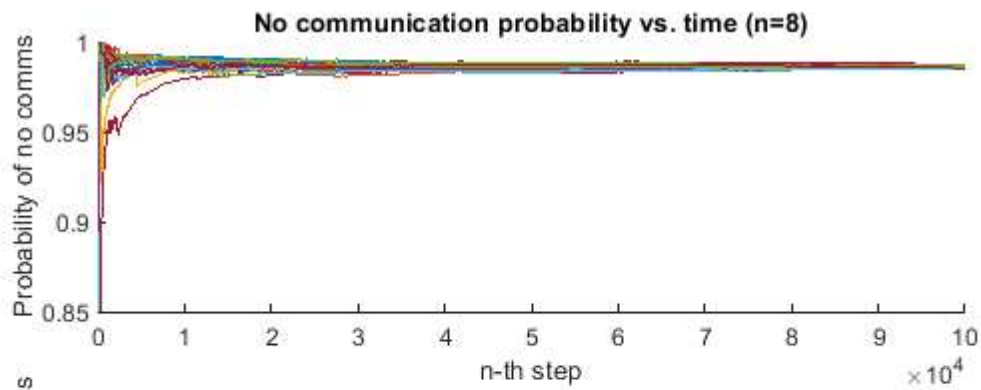
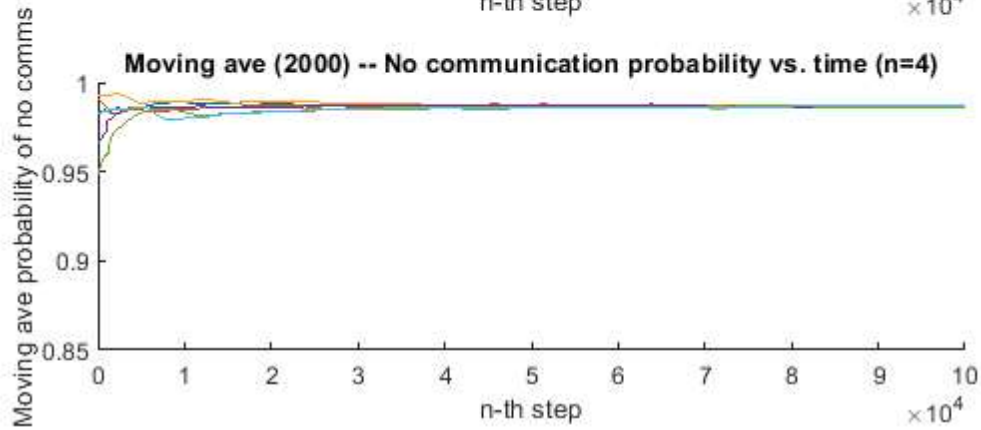
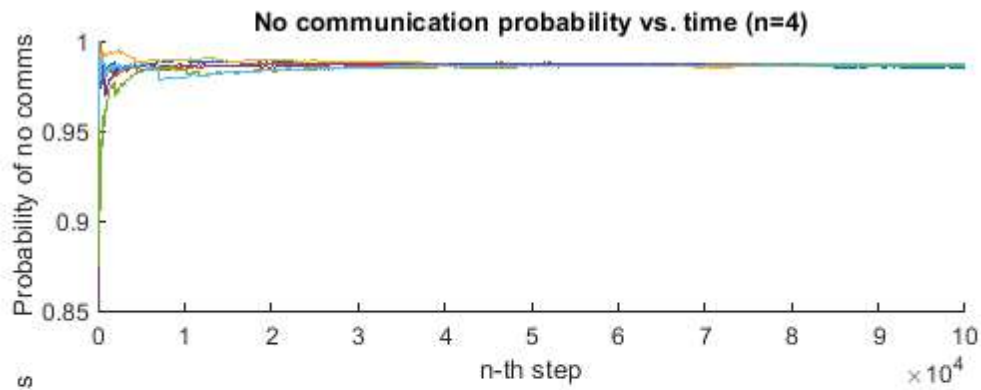
#### 4, 8, 16, 32 Agent Inf Horizon Test in 20x40 Box

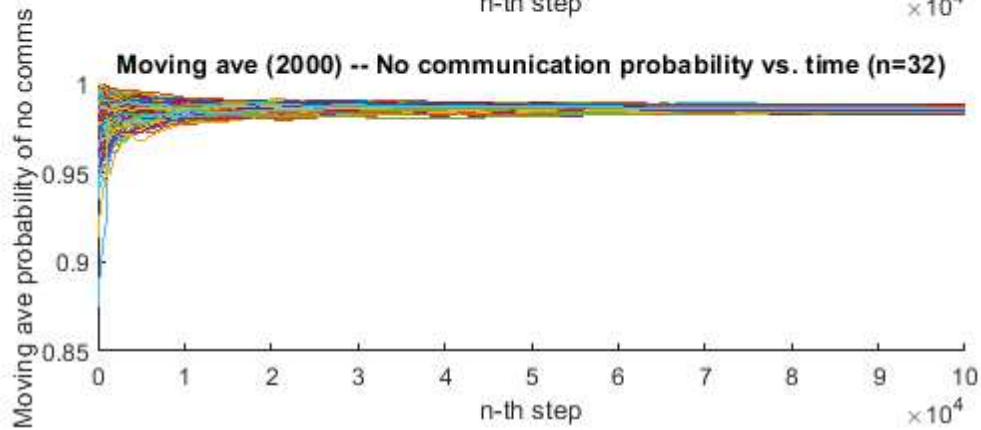
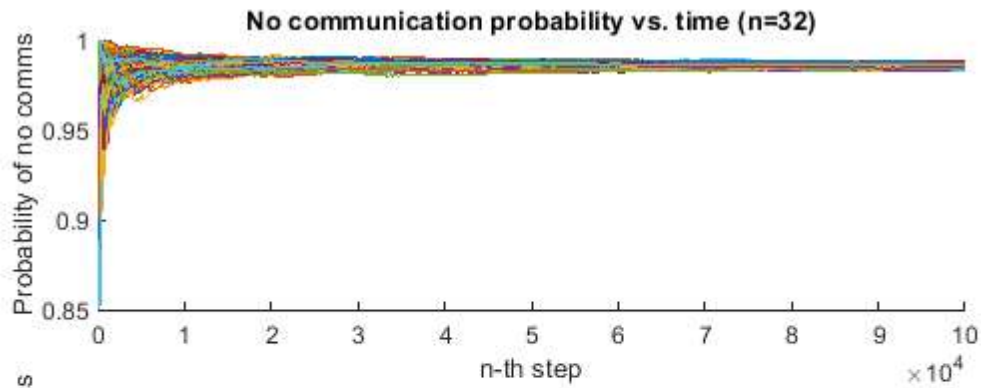
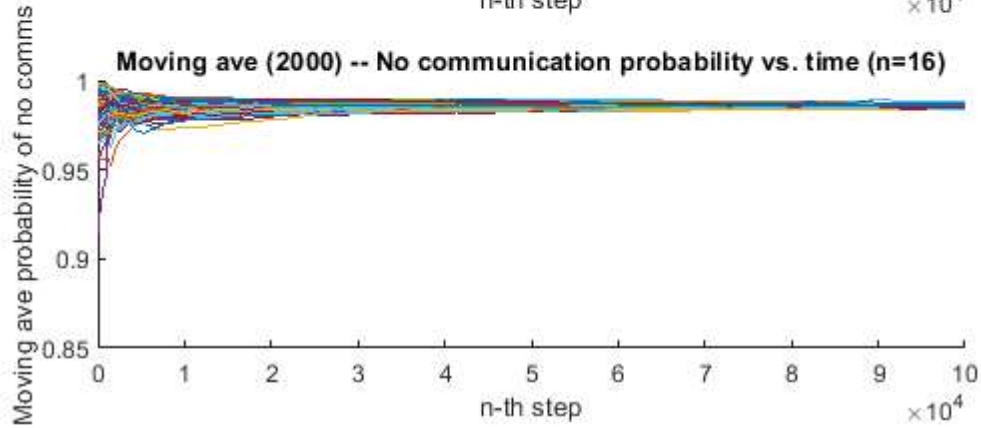
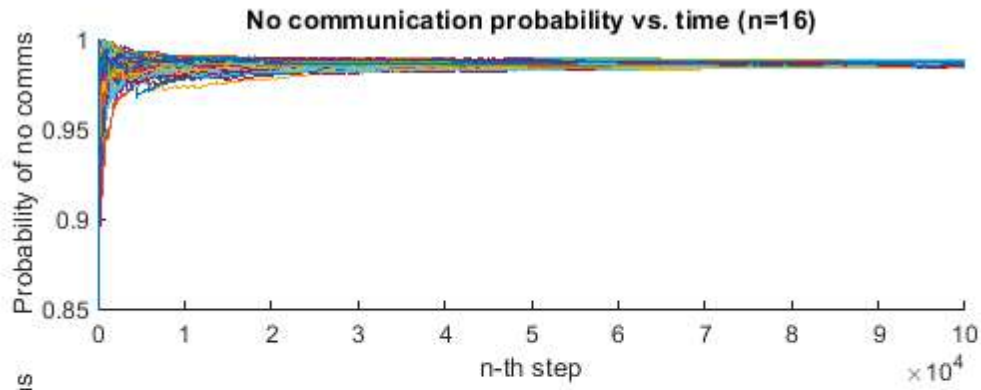
Results: 4-Agent case: converges to [0.984, 0.985] 8-Agent case: converges to [0.983, 0.987] 16-Agent case: converges to [0.983, 0.988] 32-Agent case: converges to [0.983, 0.988]

```
bounds = [0 40 40 0; 0 0 20 20];

a.steadyStateCommProb(bounds, 4, 100000, 1);           % bounds, numAgents, N, sim_itr
a.steadyStateCommProb(bounds, 8, 100000, 1);
a.steadyStateCommProb(bounds, 16, 100000, 1);
a.steadyStateCommProb(bounds, 32, 100000, 1);

SimAnalysis.clearEmptyFigs();                           % closes figures without titles (for empty ones!)
```





#### 4, 8, 16, 32 Agent Inf Horizon Test in 20x80 Box

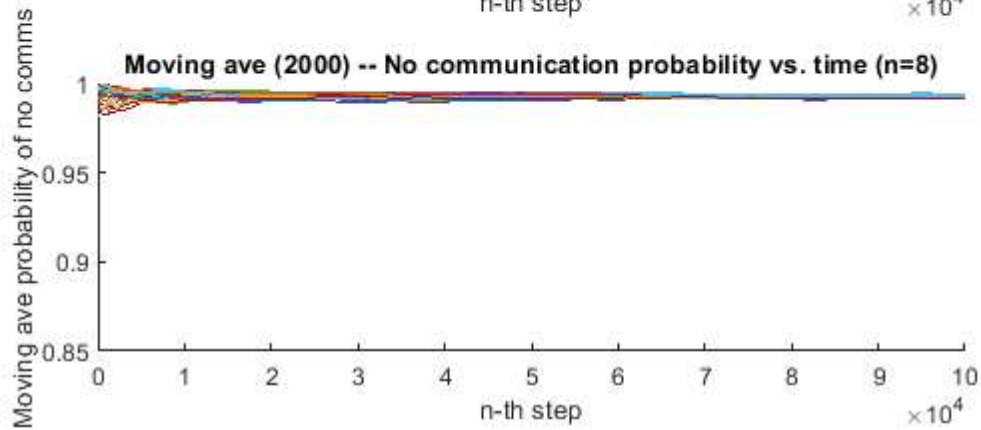
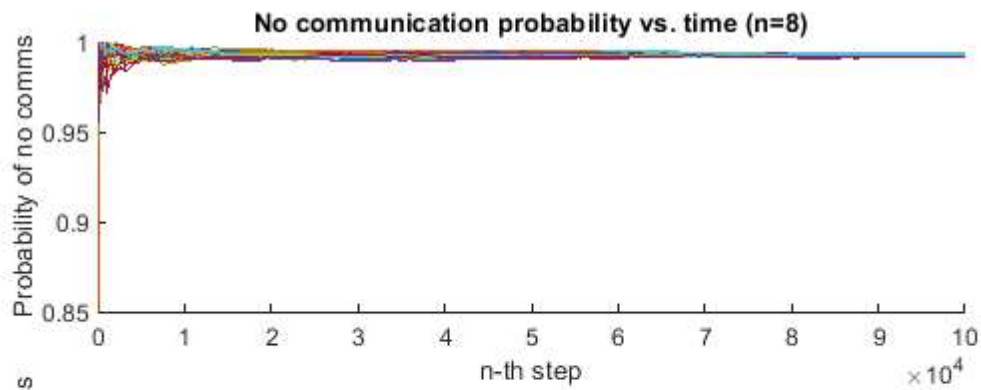
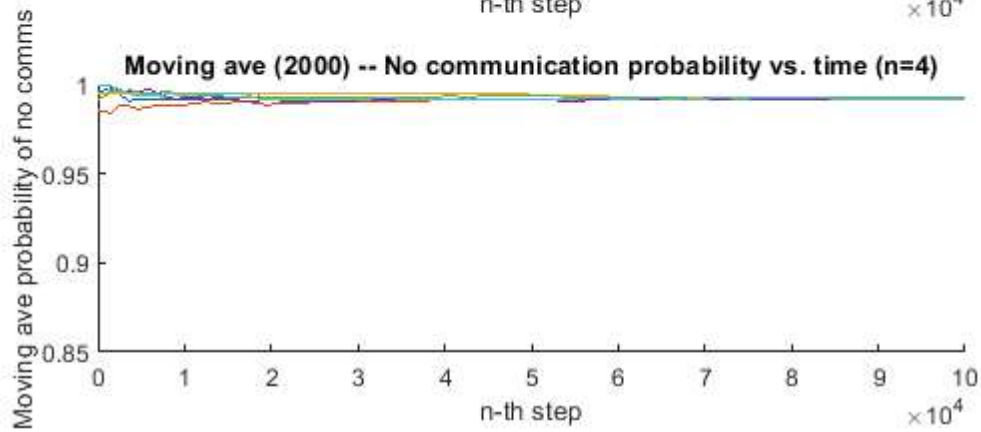
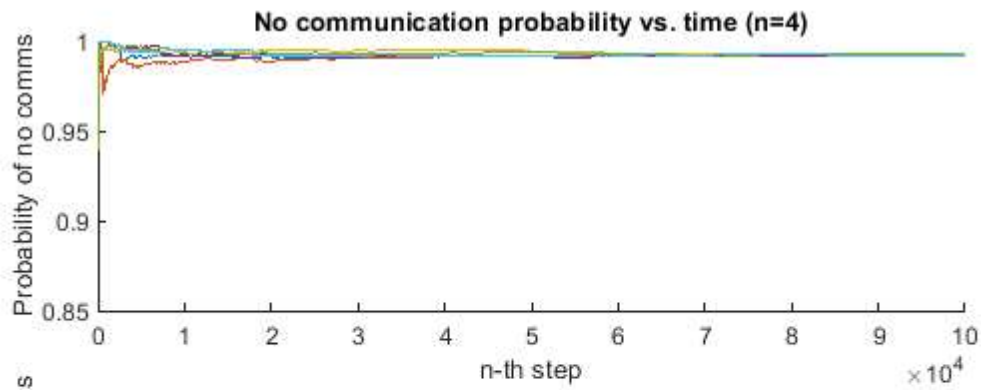
Results: 4-Agent case: converges to [0.991, 0.992] 8-Agent case: converges to [0.991, 0.993] 16-Agent case: converges to [0.990, 0.994] 32-Agent case: converges to [0.988, 0.994]

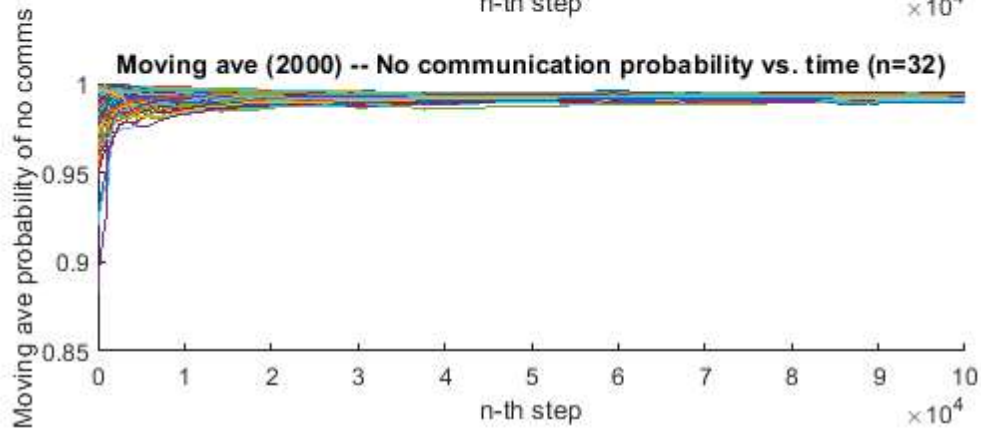
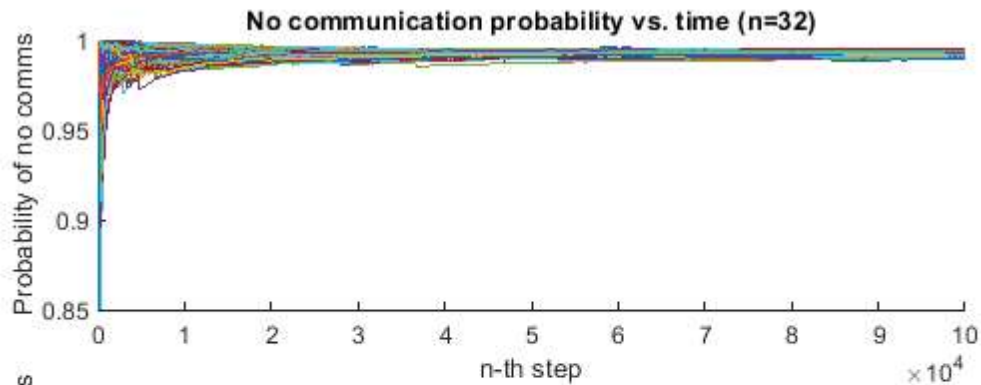
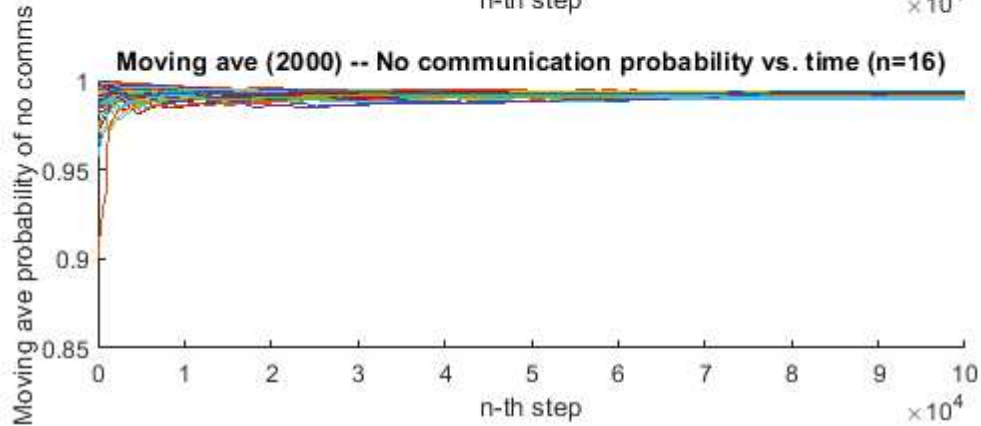
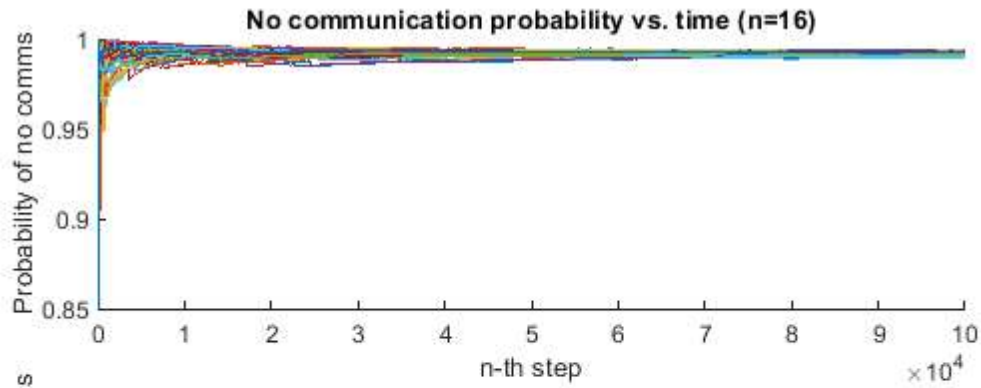
```
bounds = [0 80 80 0; 0 0 20 20];

a.steadyStateCommProb(bounds, 4, 100000, 1);           % bounds, numAgents, N, sim_itr
a.steadyStateCommProb(bounds, 8, 100000, 1);
a.steadyStateCommProb(bounds, 16, 100000, 1);
a.steadyStateCommProb(bounds, 32, 100000, 1);

SimAnalysis.clearEmptyFigs();
```







## General Analysis -- 240 path length, multi-start simulations

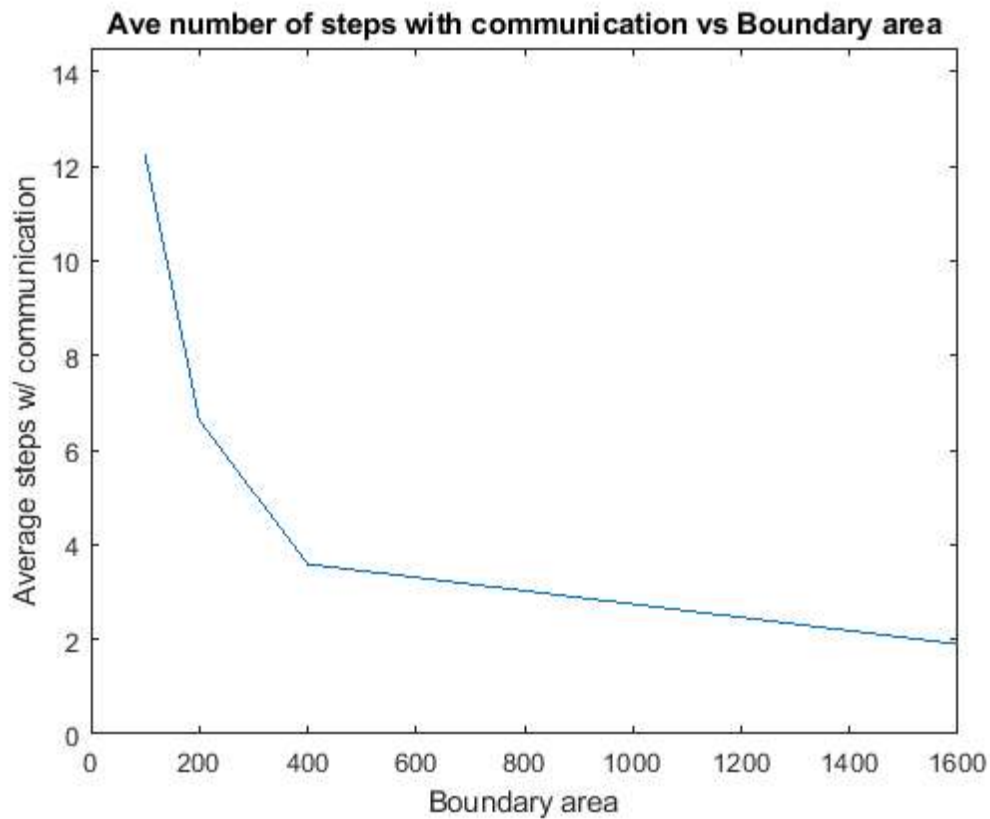
Analysis: For axis-aligned rectangular boundaries the average number of steps with communication drops off sharply as the boundary area increases, but seems stationary (wrt the mean) to the number of agents. It should be noted that the std. deviation of the average number of steps in communication appears to increase as the number of agents increases. We can also note that the number of multi-start iterations with at least one agent pairing experiencing no communications increases sharply from the 20x10

bounds to 20x40 and beyond. We know that any iteration with two agents experiencing no communication results in an un-connected graph, which needs to be avoided. Accurately describing when this occurs is a target for future work.

```
xgen = [100, 200, 400, 1600];
ygen = [12.245, 6.64, 3.59, 1.90];

figure()
plot(xgen, ygen)
xlabel("Boundary area")
ylabel("Average steps w/ communication")
title("Ave number of steps with communication vs Boundary area")

ylim([0 14.5])
```



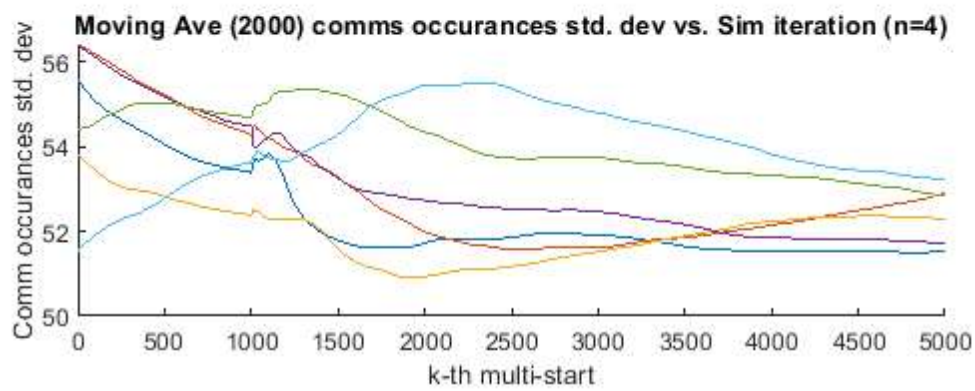
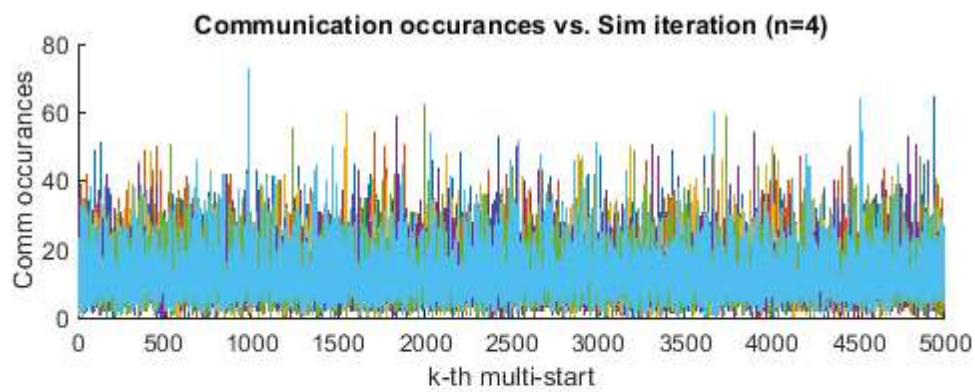
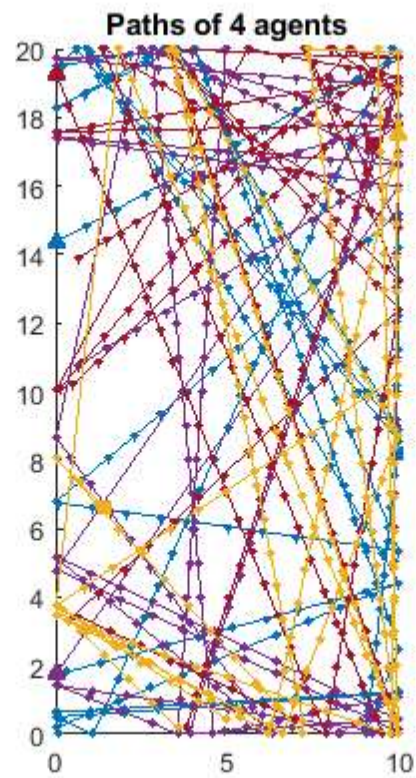
#### 4, 8, 16, 32 Agent Multi-Start Test in 20x10 Box

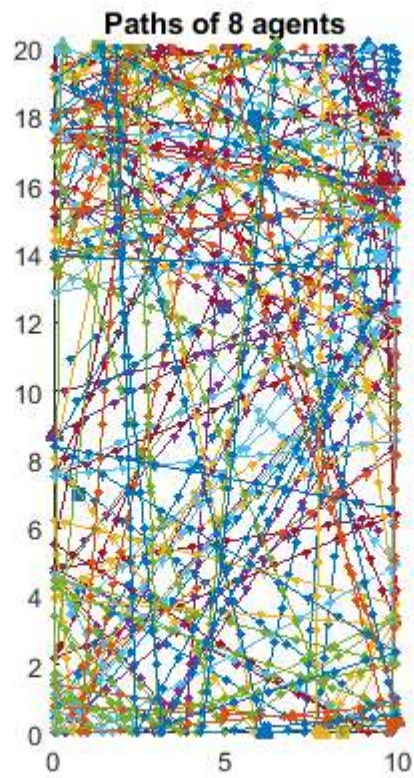
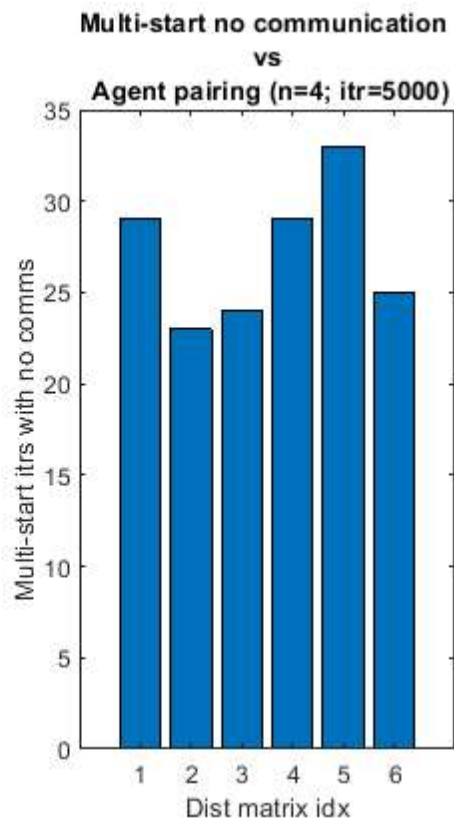
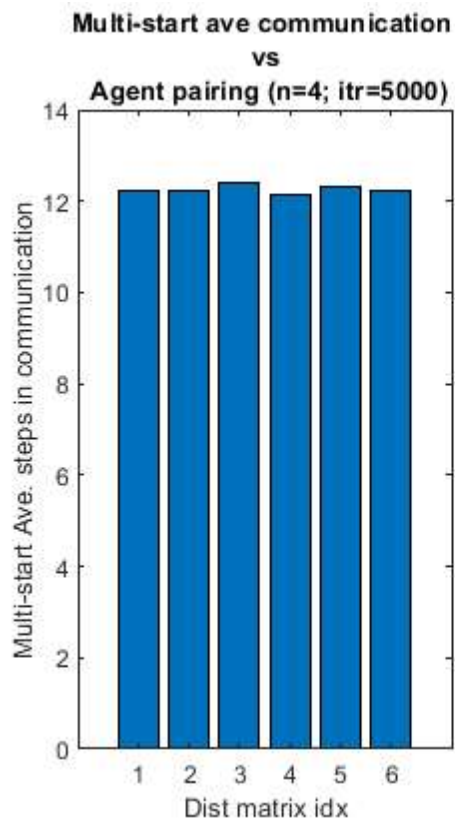
Results: 4-Agent case: ave comms [12.19, 12.46]; no comms itrs [21, 32] 8-Agent case: ave comms [11.93, 12.44]; no comms itrs [16, 36] 16-Agent case: ave comms [12.03, 12.46]; no comms itrs [14, 43] 32-Agent case: ave comms [11.86, 12.48]; no comms itrs [14, 41]

```
bounds = [0 10 10 0; 0 0 20 20];

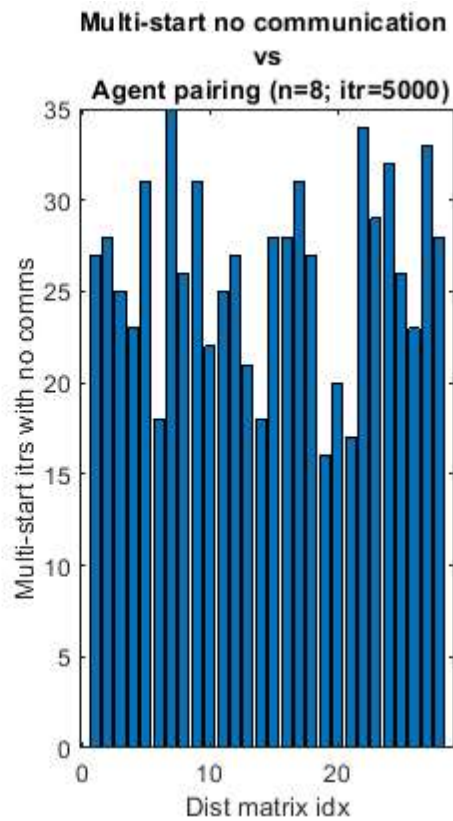
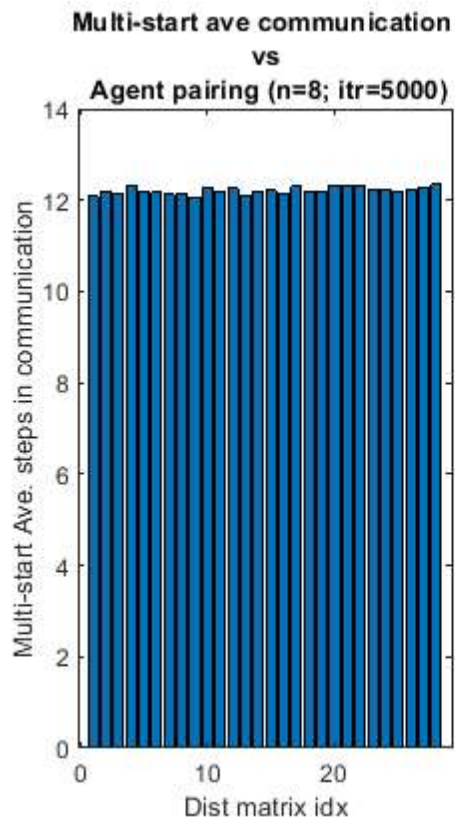
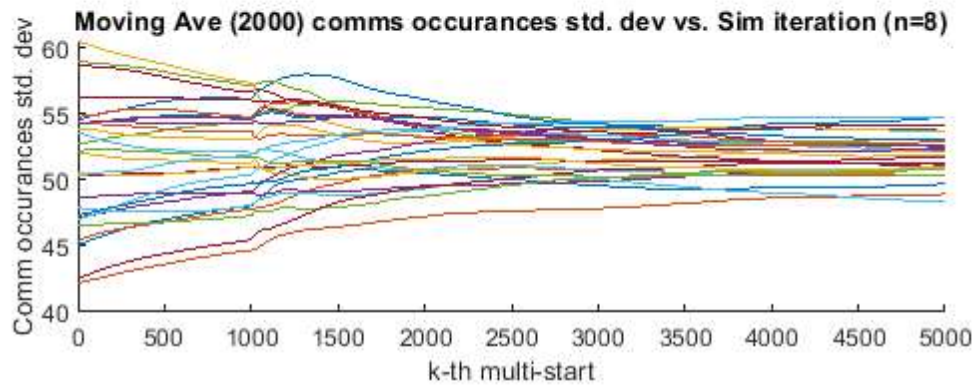
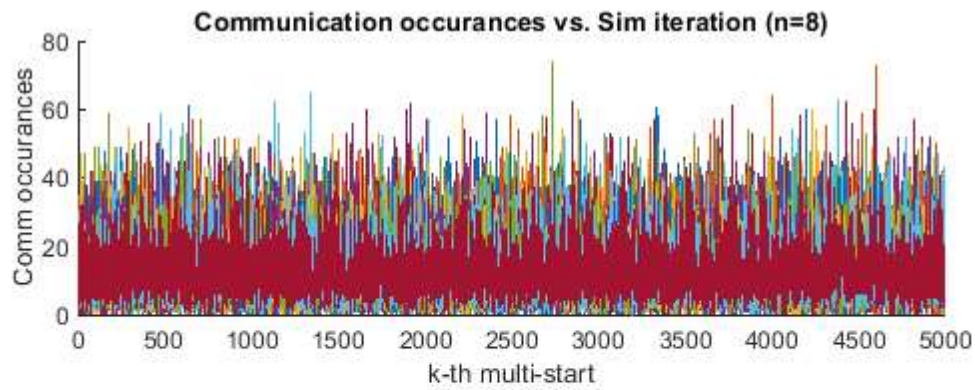
a.multiStartData(bounds, 4, 240, 5000);
a.multiStartData(bounds, 8, 240, 5000);
a.multiStartData(bounds, 16, 240, 5000);
a.multiStartData(bounds, 32, 240, 5000);

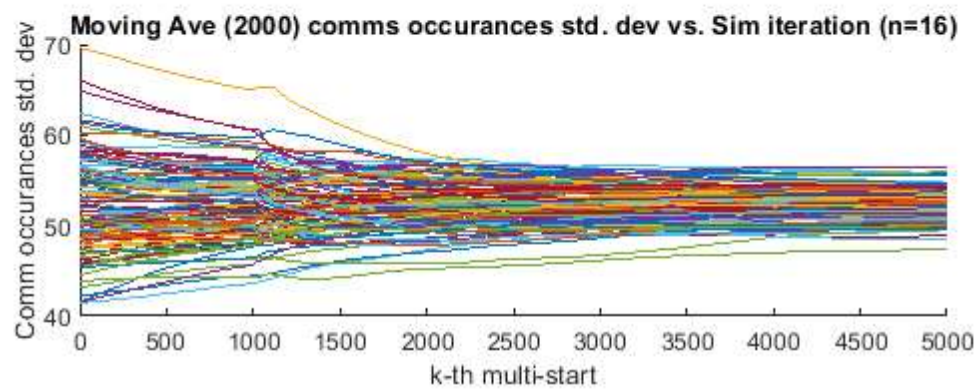
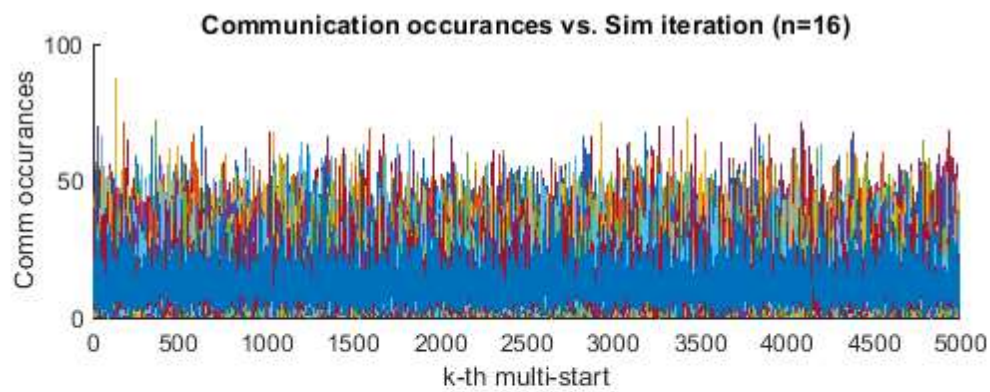
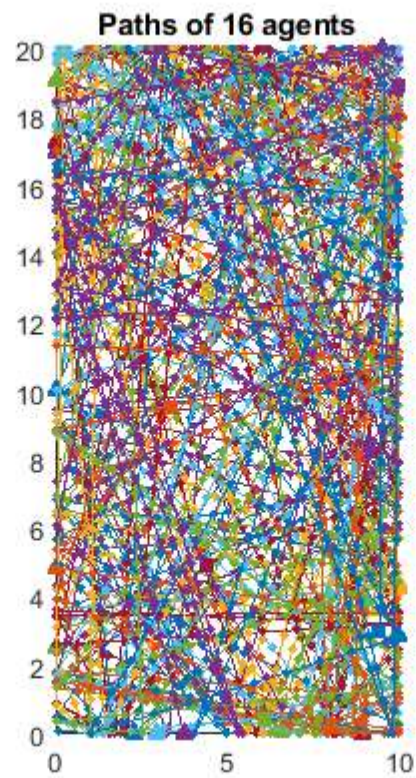
SimAnalysis.clearEmptyFigs();
```

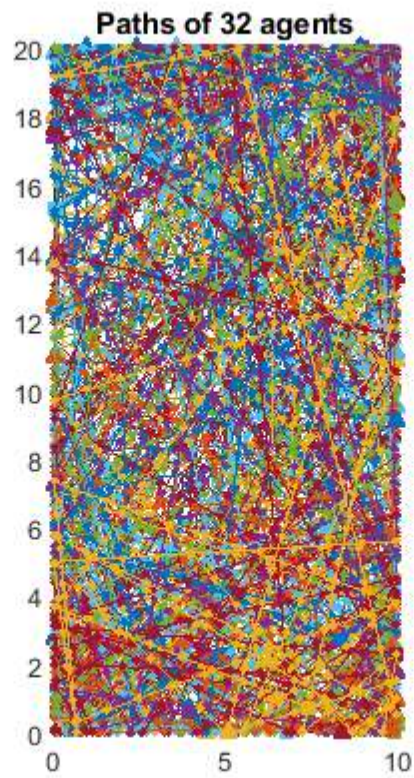
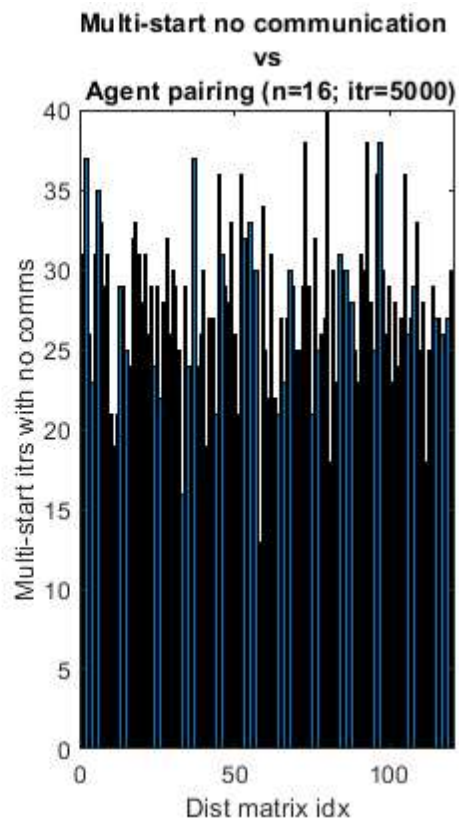
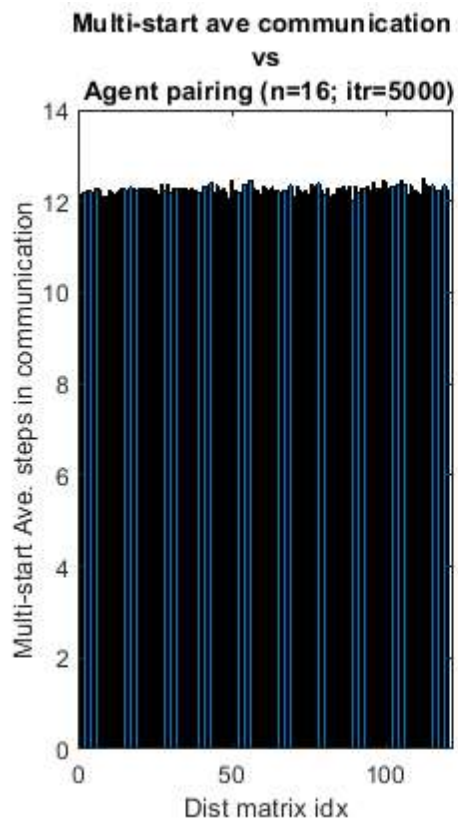




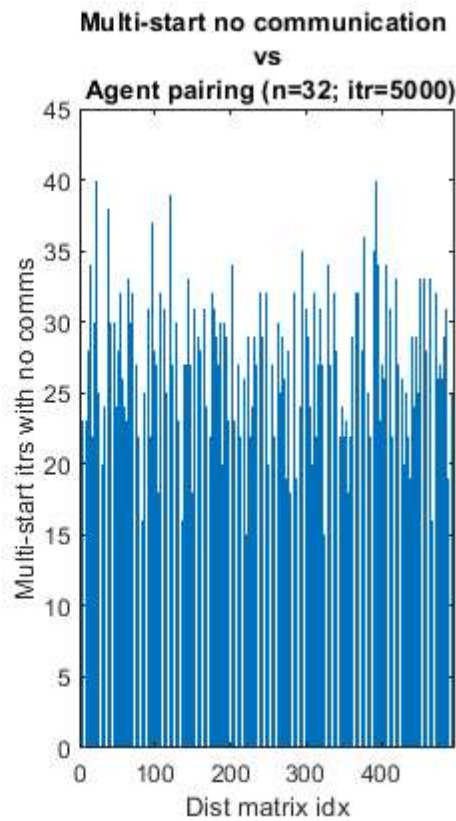
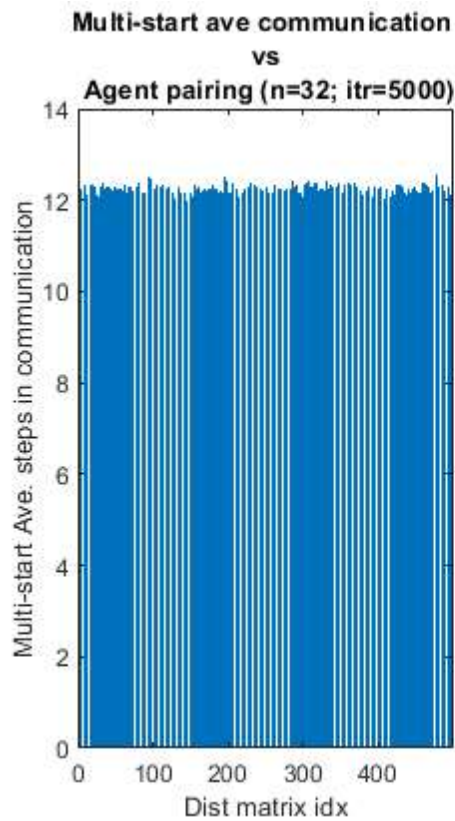
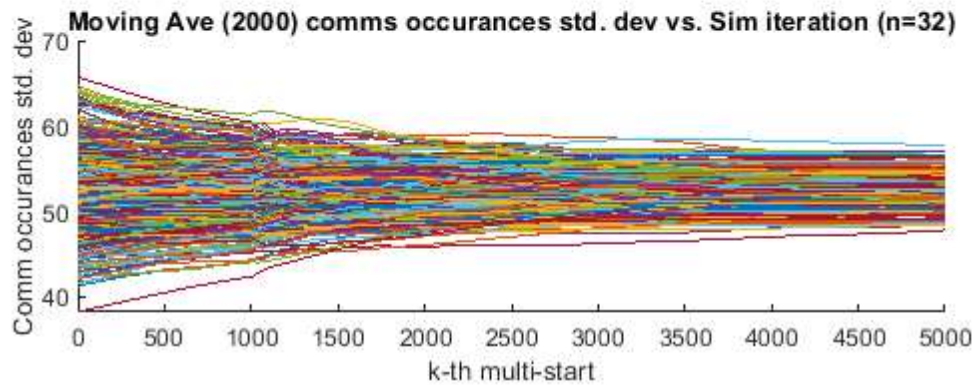
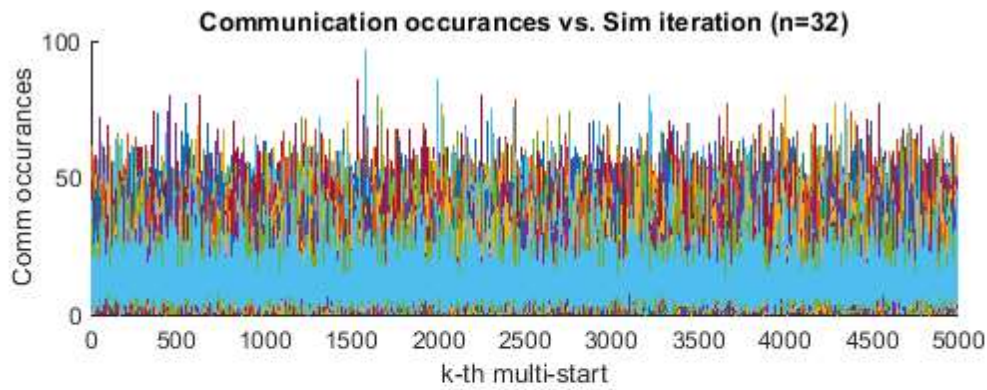












#### 4, 8, 16, 32 Agent Multi-Start Test in 20x40 Box

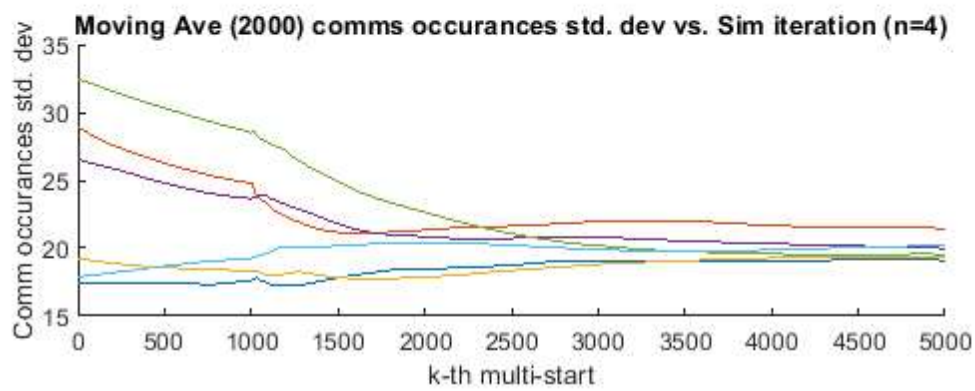
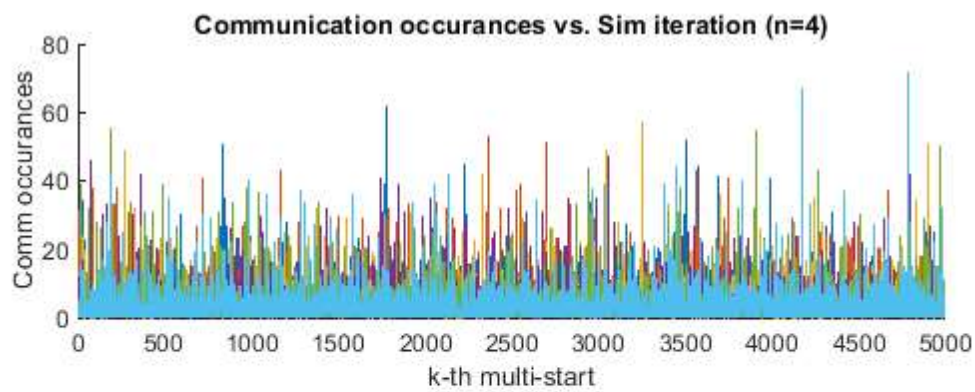
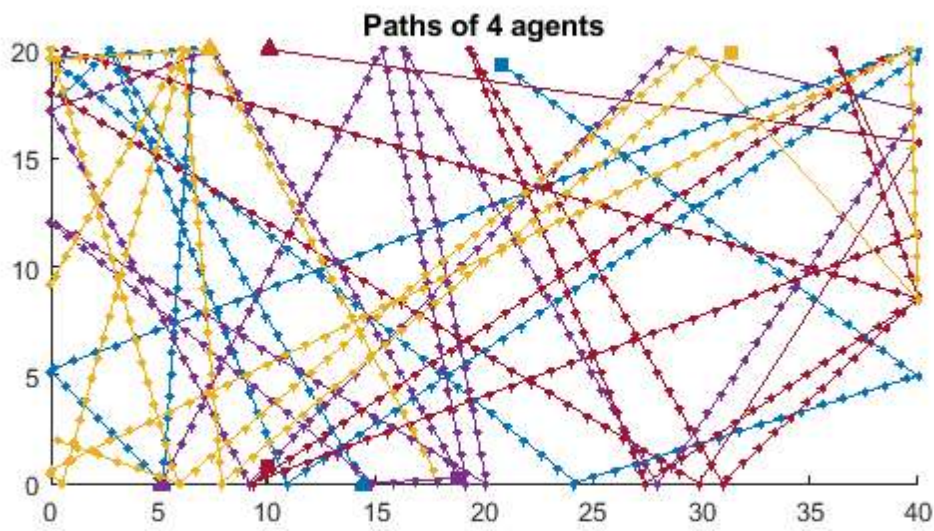
Results: 4-Agent case: ave comms [3.50, 3.68]; no comms itr [1205, 1241] 8-Agent case: ave comms [3.46, 3.75]; no comms itr [1196, 1278] 16-Agent case: ave comms [3.33, 3.71]; no comms itr [1152, 1304] 32-Agent case: ave comms [3.43, 3.76]; no comms itr [1152, 1320]

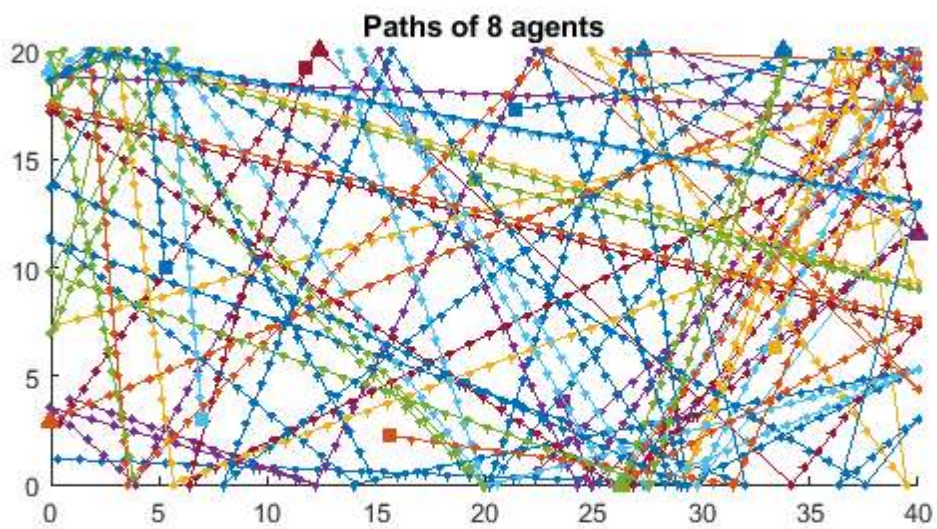
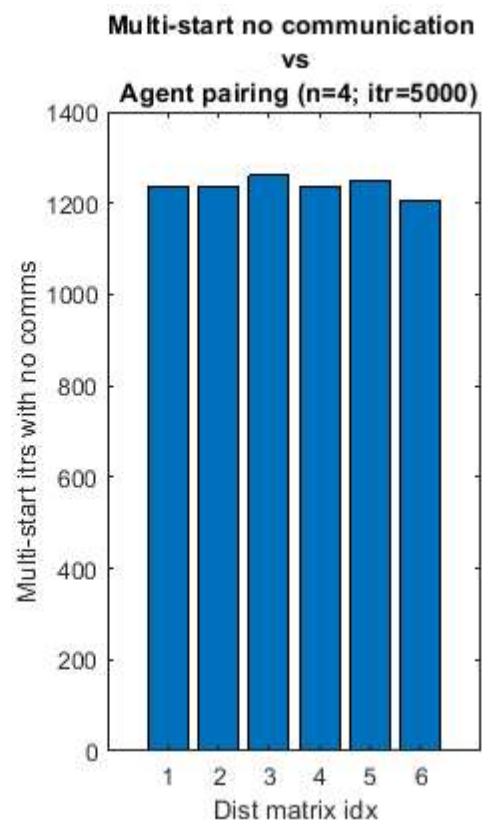
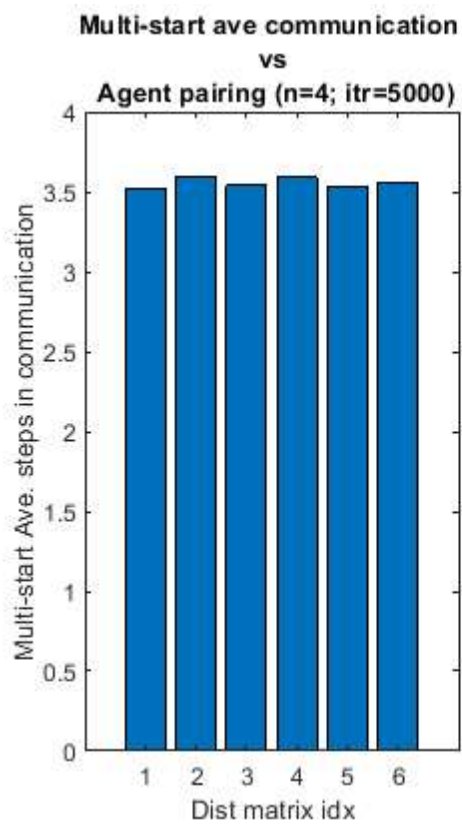
```
bounds = [0 40 40 0; 0 0 20 20];
```

```
a.multiStartData(bounds, 4, 240, 5000);  
a.multiStartData(bounds, 8, 240, 5000);  
a.multiStartData(bounds, 16, 240, 5000);  
a.multiStartData(bounds, 32, 240, 5000);
```

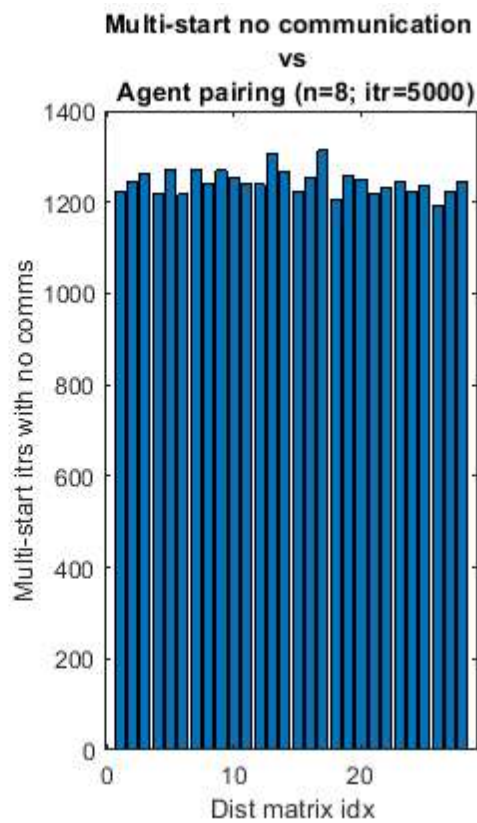
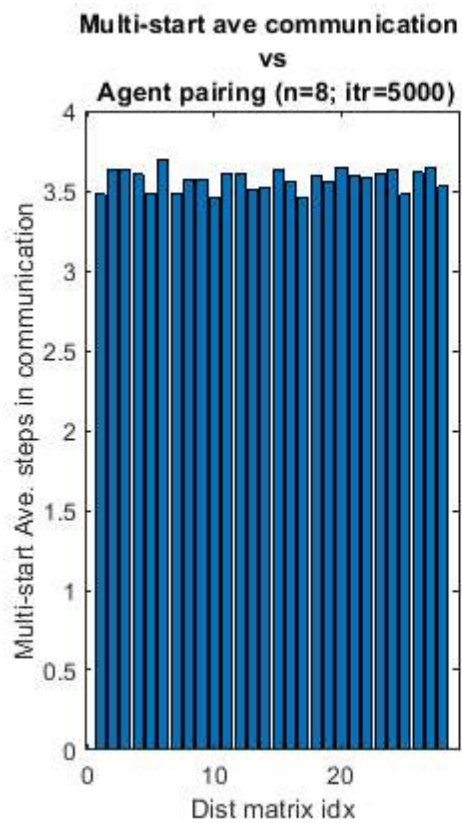
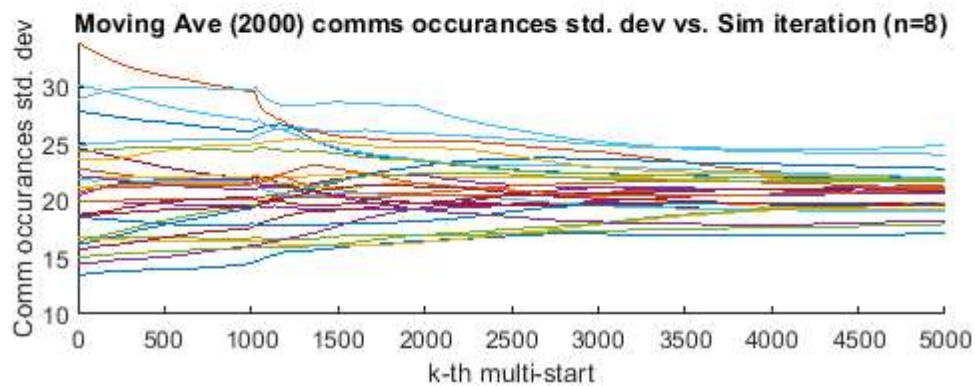
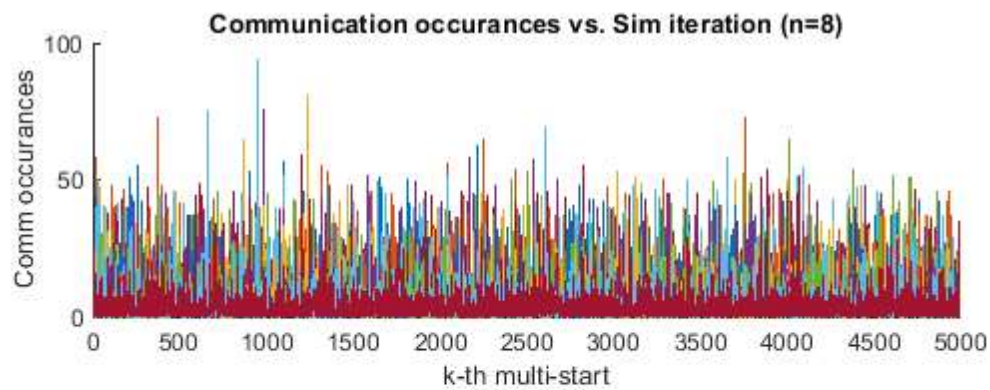
```
% bounds, numAgents, N, sim_its
```

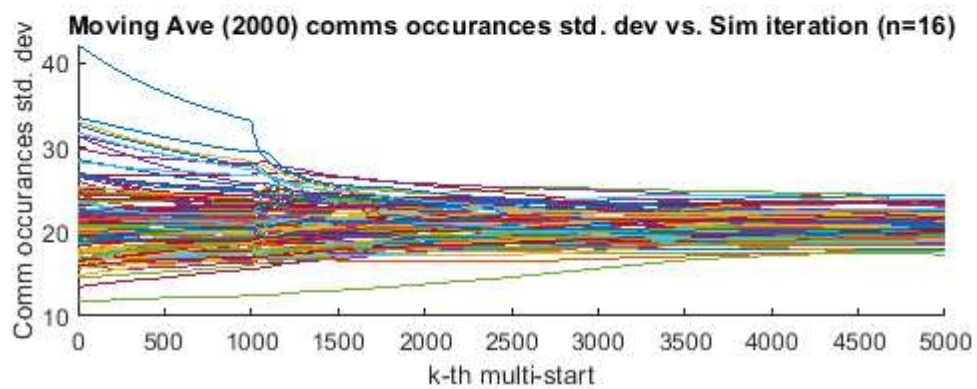
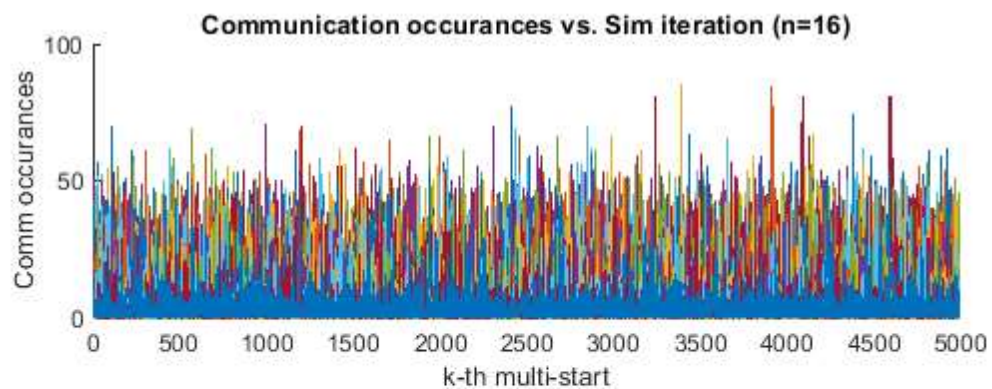
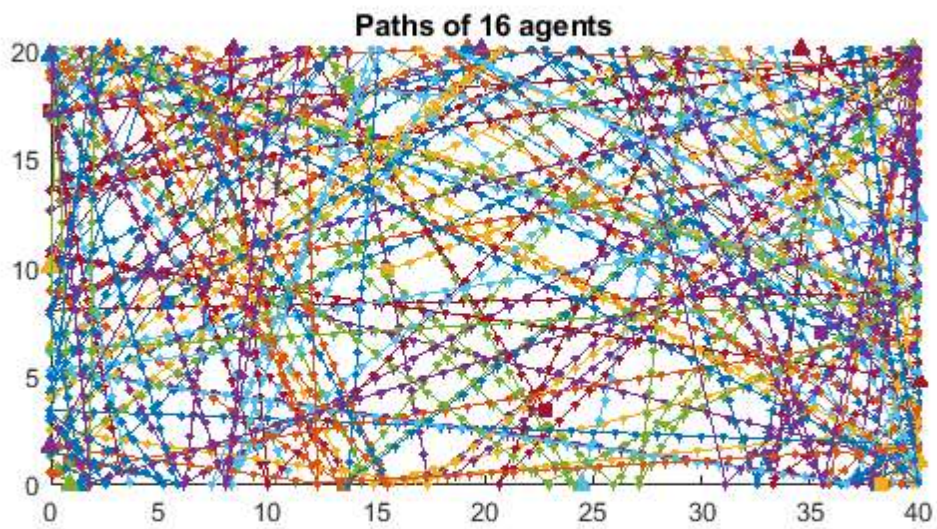
```
SimAnalysis.clearEmptyFigs();
```

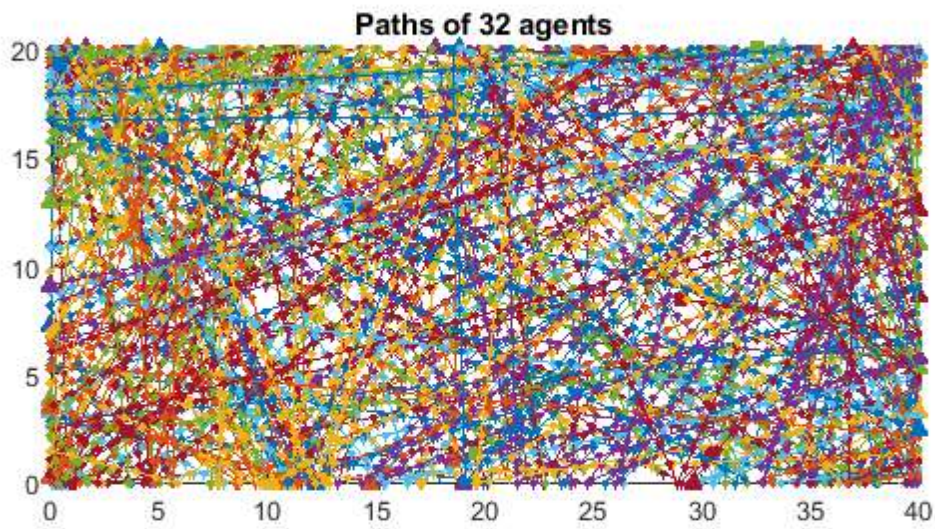
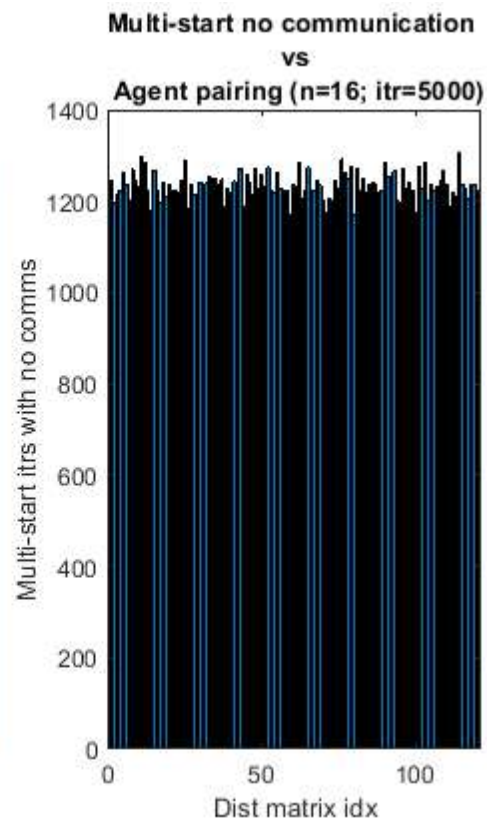
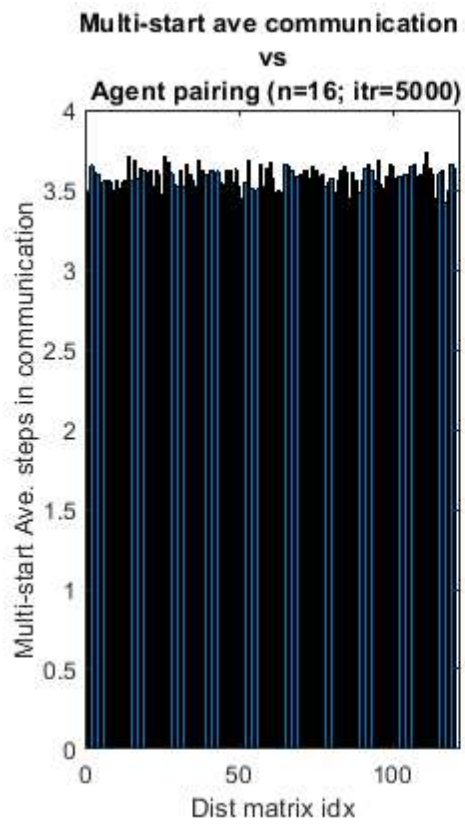




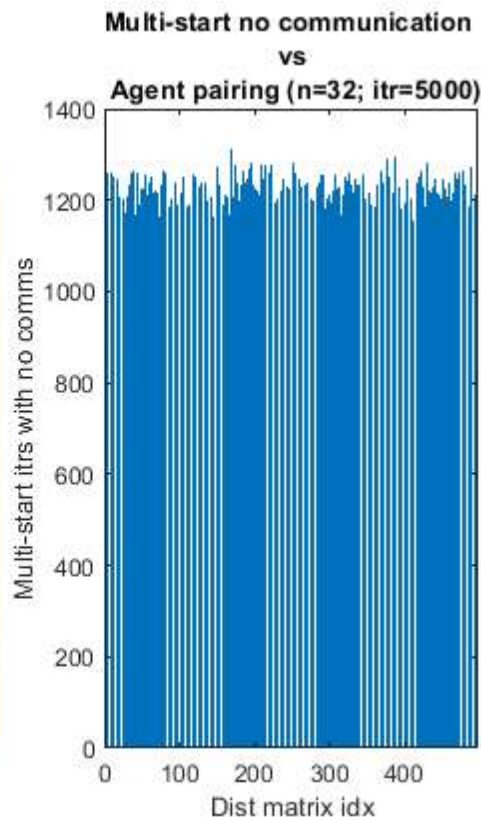
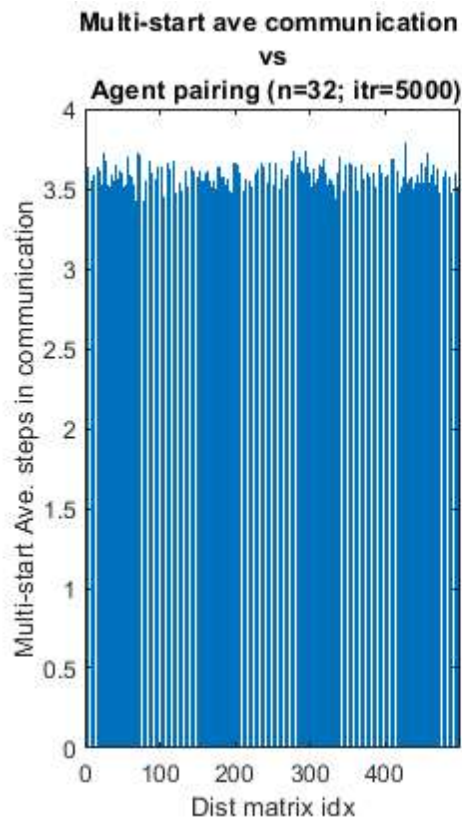
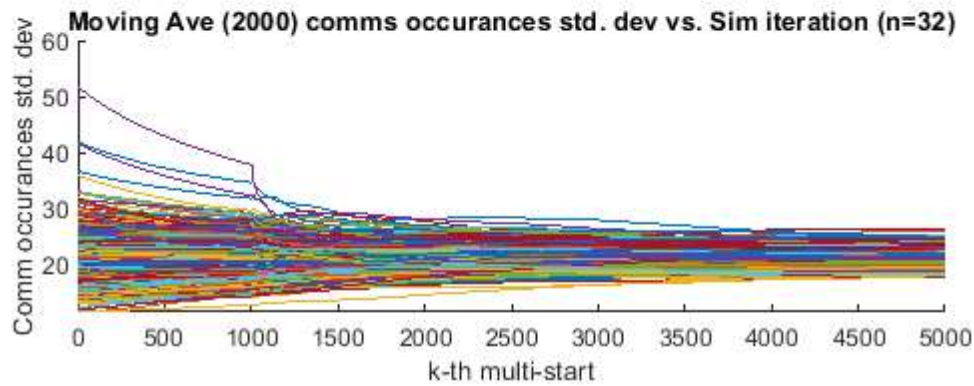
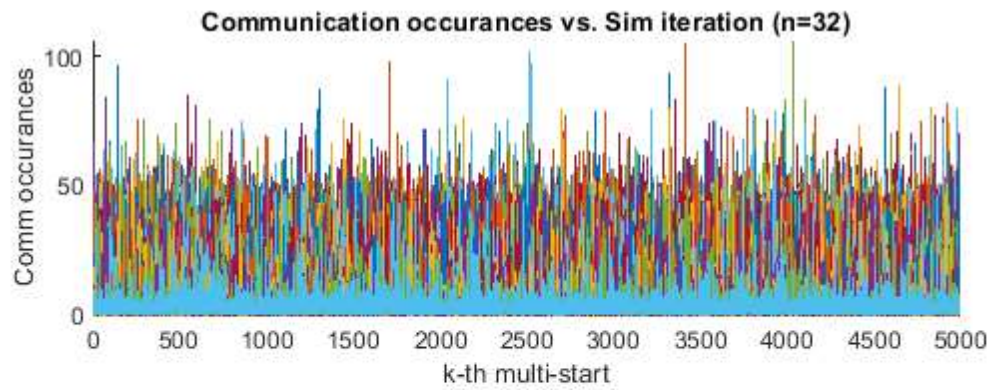












#### 4, 8, 16, 32 Agent Multi-Start Test in 20x80 Box

Results: 4-Agent case: ave comms [1.86, 2.00]; no comms itr [2479, 2522] 8-Agent case: ave comms [1.80, 1.97]; no comms itr [2453, 2558] 16-Agent case: ave comms [1.76, 2.05]; no comms itr [2438, 2589] 32-Agent case: ave comms [1.73, 2.07]; no comms itr [2430, 2651]



```
bounds = [0 80 80 0; 0 0 20 20];

a.multiStartData(bounds, 4, 240, 5000);           % bounds, numAgents, N, sim_its
a.multiStartData(bounds, 8, 240, 5000);
a.multiStartData(bounds, 16, 240, 5000);
a.multiStartData(bounds, 32, 240, 5000);

SimAnalysis.clearEmptyFigs();

disp('Done')
```

Done

