Introduction

• Why multi-robot system?
  • Better global system performance
  • More abilities
  • More robustness
  • Lower cost

• What is multi-robot coordination?

  A team of robots interacting with others to reach a common goal.
Introduction

• Why building efficient multi-robot coordination is difficult?

  Most problems are **NP-hard**:  
  • Multi-robot Task Allocation  
    • Resource allocation  
    • Exploration

  • Multi-robot Motion Planning  
    • Routing  
    • Trajectory planning
Introduction

• How to reduce the difficulty?
  • Reduce the environment space
  • Reduce the action space
  • Estimate a finite horizon of future
Outline

• Known Environment with Global Communication:
  • Intelligent In-Orchard Bin-Managing System For Tree Fruit Harvest
  • Multi-robot Routing for Dynamic Information Gathering

• Unknown Environment with Limited Communication:
  • Multi-UAV Explore, Map, and Search in Unknown Environments
Intelligent In-Orchard Bin-Managing System for Tree Fruit Harvest

(aka. the Bin-dog project)
Motivation

- High labor demand of tree fruit (apple) harvest
- Low productivity with inefficient bin management

Apple harvest in Prosser, WA.
Main Goal

*To develop an intelligent bin-managing system supported by a robotic self-propelled fruit bin carrier*

“Bin-dog”, designed by Center for Precision & Automated Agriculture Systems, Washington State University
Simulation Environment

- 10 trees x 5 lanes
- Workers cost 2 steps to finish one tree
- Robots move 1 step per time when carrying a full bin, 2 steps otherwise.
Simulation Setup

• Groups of workers are initialized in the beginning of the lanes.

• No more apples?
  • Workers move to a new location.
  • Workers request a new bin.

• Robots choose **which bin to pick up**.
  • Robots wait if the target bin is not full yet.

• Robots choose **where to carry a new bin to**.
Algorithm: Baseline (Naive Greedy)

- Greedy, no coordination.
- Choose the closest full bin.
- Choose the earliest requested location.
- Choose the bin that will be filled faster.
  - Wait if the target bin is not full.
- Priority: other robots cannot see a chosen bin/request.

Not efficient!
Algorithm: Auction-based

• Robots coordinate through auction.
  • Each robot makes plans to pick up a bin.

• Plan cost:
  \[ C = t_T + t_W \]
  • \( t_T \): the time required to reach target bin.
  • \( t_W \): the time required to wait for target bin to be full.

• The one with least cost wins the task
Simulation (Auction-based)
Results

Total steps cost to finish all the tasks (80 bins)

Lower is better

Number of bins retrieved after 150 time steps

Higher is better
Extension

“Bin-dog” is interesting, but...

- Visit multiple points of interest one time?
- How to balance the workload between robots?
- Workers work in different speed?
Multi-Robot Routing for Information Gathering
Introduction

Goal

Coordinate a team of robot to retrieve resources from a number of resource collectors in a static environment.
Introduction

Informative map
• 8-connected 2D grid
• No obstacles
• Each cell has a certain amount of information
Introduction

Collectors:

- Collect information from each cell

- Move to a neighbor cell when finish

- Individual collecting rate

- Limited capacity, pause when full
Introduction

Robots

• Visit collectors to retrieve information

• Consistent speed

• Different base stations

• Limited capacity, go back if full

Robot can retrieve a portion of information from each collector
Problem Formulation

- **Given:**
  - $n$ collectors
  - $m$ robots

- **Objective Function**

$$\arg\max (\forall p \downarrow i \in \Psi \ Info(R\downarrow i \ldots m) - Idle(T\downarrow i \ldots n))$$
Sub-problems

• Two sub-problems
  • Multi-robot Task allocation:
    • which collector should be assigned to which robot?
  • Multi-robot Motion Planning:
    • Visit the goal collectors in which order?
    • Retrieve how much information from a specific collector?
Algorithm: Sequential Auction with Greedy Path Planning

• **Task Allocation:** Sequential Auction
  - Sort the unassigned tasks (collectors) by *urgency*:
    \[ U_{\downarrow i} = (\text{collecting rate} + \text{current fullness})/\text{distance} \]
  - Auction tasks based on urgency
  - Cost of each robot:
    \[ C_{\downarrow i} = \text{total travel distance of current tasks} + \text{distance from last task to the auctioning task} \]
  - Reassign if any robot idle
Demo: Sequential Auction with Greedy Path Planning

Be *exact* full after visit all assigned collectors

\[ U_{↓1} > U_{↓2} \]
\[ T = 11 \]

Information Collected Estimation:
\[ I = (r_{↓1} + r_{↓2} \cdot T) + CUR \]

Information Retrieve for collector 1:
\[ a_{↓1} = (r_{↓1} \cdot T + CUR_{↓1}) \]
\[ a_{↓2} = (r_{↓2} \cdot T + CUR_{↓1}) \]

\[ a_{↓1} + a_{↓2} = \text{robotCapacity} \]
Algorithm: Sequential Auction with Greedy Path Planning

- **Motion Planning**: Greedy
- **Path Planning**: 1-horizon greedy. Visit the most urgent collector first.

- **Information Gathering Planning**
  - $T$ : time required to travel the path.

- Estimated information collected by collector $i$:

\[
\Delta \downarrow i = \text{content} \downarrow i + T \times f \downarrow i
\]

- Amount to take (Try to be exact full after visit all assigned collectors):

\[
a \downarrow i = \Delta \downarrow i / \sum_{j=1}^{\mid P \mid} \Delta \downarrow j \times \text{RobotCapacity}
\]
Algorithm: Distributed Sampling with RH-based Path Planning

- **Task allocation**: distributed sampling.
  - **Goal**: evenly distribute the workload to the robots
Demo: Distributed Sampling with RH-based Path Planning

Create fully connected Graph $G$

$U_{\downarrow 1} = 3$
$U_{\downarrow 2} = 4$
$U_{\downarrow 3} = 2$
$U_{\downarrow 4} = 5$

algorithm modified from (Kim and Shell, 2014)
Demo: Distributed Sampling with RH-based Path Planning

Random part it into $m$ parts.

Cost(P1) = sum($U\downarrow_{1,3,4}$ ) + sum($E\downarrow_{13,14,34}$ ) = 19

Cost(P2) = 4

Diff(P1, P2) = 15
Demo: Distributed Sampling with RH-based Path Planning

Balance by move the boundary nodes.

\[
\begin{align*}
\text{Cost}(P1) &= 10 \\
\text{Cost}(P2) &= 9 \\
\text{Diff}(P1, P2) &= 1 < \text{previous Diff15}
\end{align*}
\]

A better partition!
Demo: Distributed Sampling with RH-based Path Planning

Balance by move the boundary nodes.

Cost(P1) = 2
Cost(P2) = 23

Diff(P1, P2) = 21 > 1

previous Diff
Demo: Distributed Sampling with RH-based Path Planning

Continue until converged

\[ U_{\downarrow 1} = 3 \]
\[ U_{\downarrow 2} = 4 \]
\[ U_{\downarrow 3} = 2 \]
\[ U_{\downarrow 4} = 5 \]
Workload Partition

Steps:

- Create a fully connected graph $G = (V, E)$
- $V$ is the set of all nodes (collectors), $E$ is the set of edges.
- Optimization problem:

$$\text{arg min}_{(G_i, G_j)} \left[ \text{max}_{i \in [1,m]} \left( \sum_{\forall a,b} (w_{va}^i + w_{eb}^i) \right) - \text{min}_{j \in [1,m]} \left( \sum_{\forall c,d} (w_{vc}^j + w_{eb}^j) \right) \right].$$

Time required to reach capacity

Collector’s fill rate

$$w_{vl} = -t_{vl} + f_{vl}$$

$$w_{le} = \text{length}(e_{lb})$$
Proposed Algorithm: Distributed Sampling with RH-based Path Planning

• Routing
  • Receding Horizon-based routing.
  • Look ahead \( h \) steps.

• Preferences of visiting a node \( v_j \):

\[
c_{\downarrow i} = \frac{U_{\downarrow i}}{d_{\downarrow i}}
\]

• \( d_i = \text{dist}(R, v_i) \) if \( v_j \) is the first node in the path
• \( d_i = \text{dist}(v_j, v_i) \) otherwise
Receding Horizon Path Planning

Find which collector to visit first

3-horizon planning

Route 1
Route 2
...
Route 6
Proposed Algorithm: Distributed Sampling with RH-based Path Planning

• Information Gathering Planning
  • T : time *required* to travel the path.

  • Estimated information collected by collector $i$:
    $$\Delta \downarrow i = \text{content} \downarrow i + T \times f \downarrow i$$

  • Amount to take (Try to be *exact* full after visit all assigned collectors):
    $$a \downarrow i = \Delta \downarrow i / \sum_{j=1}^{\vert P \vert} \Delta \downarrow j \times \text{RobotCapacity}$$
Simulation

• 10 x 10 cells. Each cell contains 200 units of information.
• Collectors: 2 collectors with collecting rate 5, 3 collectors with rate 2, 5 collectors with rate 1.
• Capacity: Both robots and collectors have a capacity of 100 to store information.
• Robots start from different corners.
Idle Time (1000 steps)
Information Loss (1000 steps)

Lower is better
Information Gain (1000 steps)

Higher is better

- **Distributed Sampling**
- **Sequential Auction**

Information Gain

Time Step

<table>
<thead>
<tr>
<th>Time Step</th>
<th>Information Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1000</td>
<td>7000</td>
</tr>
</tbody>
</table>
Extension

Known environment vs. Unknown environment

- Certain vs. Uncertain

Global communication vs. Limited Communication

- Centralized vs. Decentralized
Multi-UAV Explore, Map, And Search in Unknown Environments
Introduction
Goals

- An approach enable a team of UAVs simultaneously explore, map, and search in unknown environments.

- A mechanism controls the UAVs more focus on one or two sub-tasks (exploration, mapping and search).

- A communication scheme efficiently the human operators during the mission.
Problem Setup

• Environment
  • Indoor Environment with different types of rooms
  • Some rooms contain targets

![Diagram of office and lab spaces with target rooms marked (T)]
Problem Setup

• Topological Representation
  • Model the environment as a graph with rooms as nodes
UAV Model

- Limited Battery Life

- Equipped with two types of sensors:
  - Observe the room type
  - Observe the target existence
  - Both of the sensors have noise

- Limited communication (disk model)
UAV Belief State

• Room type (assume UAVs know the all types of rooms)
  For room $r$, probability of room type:
  $$P_{\downarrow r} = [P_{\downarrow t1}, P_{\downarrow t2}, ..., P_{\downarrow tn}] \ (n = |R|)$$
  $$\sum_{i} P_{\downarrow ti} = 1$$

• Target existence
  For room $r$, probability of target existence
  $$P_{\downarrow t} = [P_{\downarrow T}, P_{\downarrow F}]$$
  $$P_{\downarrow T} + P_{\downarrow F} = 1$$
Priors

- Priors provide the probability of a specific type room contains a target.

<table>
<thead>
<tr>
<th></th>
<th>Office</th>
<th>Lab</th>
<th>Corridor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>0.6</td>
<td>0.7</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Bayesian Update

- Bayesian Update
  - $PR^{t+1} T = \frac{P(T|R)}{P(T)} P(R^{t})$
  - $PT^{t+1} R = \frac{P(R|T)}{P(R)} P(T^{t})$
- The posteriors become the new priors.
Reward Function

Robots choose one of the three actions:

• Visit a neighbor node (exploring)

\[ I_{\downarrow E} = I \alpha \uparrow \text{visited} \quad (I \text{ is the utility of first visit, } 0 < \alpha < 1) \]

• Observe for room type (mapping)

\[ I_{\downarrow m} = -\sum_{r \in R} P_{\uparrow r} \log P_{\downarrow r} \]

• Observe for target (search)

\[ I_{\downarrow s} = T \downarrow , \quad T \in [0,1] \quad (\text{real utility gain}) \]
\[ I_{\downarrow s} = P \downarrow T , \quad P \downarrow T \in [0,1] \quad (\text{estimation}) \]
Weighted Sub-goals

Weights are set up to more focus on one or two sub-tasks

\[ \omega = [\omega \downarrow e, \omega \downarrow m, \omega \downarrow s] \]

\[ \omega \downarrow e + \omega \downarrow m + \omega \downarrow s = 1 \]

How to apply:
use \( \omega \) times the estimated reward to generate new estimated reward
Results

Exploring: 0.9 Mapping: 0.05 Search: 0.05

Exploring: 0.05 Mapping: 0.9 Search: 0.05

Exploring: 0.05 Mapping: 0.05 Search: 0.9
Communication Loss Constraint

A communication loss constraint $S$ is set up that only allows each UAV lose a valid communication link with human operators no more than $S$ steps.

Idea: Engage the human operators during the mission
Algorithm: Baseline

- Each UAV selects the action with the best estimated reward.

- Change if conflict with other UAVs with higher estimated reward for the same action.

- Have to select the goals without violate the constraint. Communicate with base station (human operators) at least every S steps.
Algorithm: ST-EMS (Steiner Tree – Explore, Map, Search)

Explorer and Relay

• Explorer: explore, map, and search the environment base on the reward functions and weights.

• Relay: retrieve the explorer’s information to base station so explorers have more freedom to fulfill the missions.
Algorithm: ST-EMS

Plan a goal location

Lose the communication with base?

Yes

Negotiate with others in its range

Anyone with lower utility or already is a relay

Yes

Relay connection built

No

Who should be relays?

Look for a relay:

October 1, 2015
Algorithm: ST-EMS

- Find relay locations: Steiner Minimum Tree with Minimum Steiner Points and bounded edge length (SMT-MSP)

- Steiner Minimum Tree

  Given a set $V$ of vertices, interconnect them by a graph of shortest length
Algorithm: ST-EMS

- **SMT-MSP**
  - Find the best relay locations
Algorithm: ST-EMS

- SMT-MSP is NP-hard (Chen et al, 2000)
- Greedy approximate (Du and Hu, 2008)
Algorithm: ST-EMS

- **Explorers Make Decision**
  - If relay needed
    - Only one time
    - Have a relay?
      - Yes: Relays Make Decision
      - No: Look for a relay
  - No relay available

- **Look for a relay**
  - Need a new relay
  - Infeasible Positions

**Decision Flowchart**

- **Relays Make Decision**
  - Have a relay?
    - Yes
    - No

**Notes**

October 1, 2015
Simulation: Environment

BS

C C C C C C C

O O O O O L L L

October 1, 2015
Results

- Total reward gain for both algorithms for 50 steps
Results

• Scalability: total reward gain for 50 steps
Results

- Reward along with different communication loss constraint
Summary of Contributions

• Known Environment with Global Communication:
  - An intelligent in orchard auction-based bin-management system
  - Dynamic information gathering with even workload distribution and RH-based routing

• Unknown Environment with Limited Communication:
  - Multi-UAV Explore, Map, and Search simultaneously with operator preferences and communication loss constraint
Conclusion

Multi-robot coordination is difficult because:

- Large state space
- Many action choices
- Dynamic environments

We approach them by:

- Estimating a finite horizon of future changes
- Each robot makes independent decisions while contributing to a common objective
- Underlying representation allows coordination
Future Work

• Improved future predictions
  • Environment, task, teammates

• Better task decomposition and allocation
  • More sophisticated partitioning, Steiner tree approximation

• Integration with human operators
  • Learning operator preferences across environments

• Implementation: orchard bin management, UAV exploration/mapping/search
Acknowledgement
Questions?