Informative Path Planning with a Human Path Constraint

Daqing Yi  Michael A. Goodrich  Kevin D. Seppi

Department of Computer Science
Brigham Young University
# Outline

## Structure

1. **Introduction**
2. **Problem definition**
   - Informative path
   - Human constraint
   - The optimization model
3. **Solution**
   - Hardness of problem
   - Backtracking heuristic
   - Anytime algorithm design
4. **Simulation**
   - Robot wingman
   - Results
5. **Summary and future work**
Human-robot collaboration

Introduction

Human

Robot

Human

Autonomous

Robot

Human

Human-robot interaction

Human-robot collaboration
Cordon and search

Introduction
Coverage model
Informative path

- Information measurement - entropy
- Maximum coverage problem
Submodularity

Informative path

\[ f(A) + f(B) \geq f(A + B) \]

Information

- search space \( S \)
- the observation of a robot \( O^X \)
- the observation of a human \( O^Y \)

\[ f(S, O^X) + f(S, O^{Yh}) \geq f(S, O^X, O^{Yh}) \]
Submodular orienteering
Informative path

Conditional mutual information

\[ I(S; O^X \mid O^{Y^h}) = H(S \mid O^{Y^h}) - H(S \mid O^X, O^{Y^h}) \]

- Entropy reduction
- Submodularity
- Chain rule

\[ I(S; O^X \mid O^{Y^h}) = \sum_{t=1}^{T} I(O_t^X; S \mid O_1^X, \cdots, O_{t-1}^X, O^{Y^h}) \]
Team role
Human constraint

- cooperative observation
- assistance and protection
Neighboring function
Human constraint

- human path \( \{y^h_1 \cdots y^h_T\} \)
- neighboring function \( N(y^h_T) \)

\[ y^h_{t+1} \]
\[ N(y^h_{t+1}) \]
\[ y^h_t \]
\[ N(y^h_t) \]
\[ y^h_{t-1} \]
\[ N(y^h_{t-1}) \]
Problem abstraction

The optimization model

Urban map

Topology graph

Multi-partite graph

Human path
The multi-partite graph

The optimization model

- time-space synchronization
- connection determined by discretized map
A pruning process
The optimization model

Reachable

Forward pruning
\[ \forall t \in \{2, \cdots T\}, \forall v \in V(t), \text{deg}^-(v) > 0 \]

Non-terminating

Backward pruning
\[ \forall t \in \{1, \cdots T - 1\}, \forall v \in V(t), \text{deg}^+(v) > 0 \]
Obstacles
The optimization model
Submodular orienteering on a multi-partite graph
The optimization problem

**Objective**: $X^* = \arg \max_X f(X)$;

**Constraint**: $|X| = T$, $x_t \in V(t), (x_t, x_{t+1}) \in E$. 
Bellman-like equation

Heuristic

\[
\hat{x}_t = \arg \max_{X_t} f(x_t \mid x_1, \ldots, x_{t-1}) + \max_{X_{t+1}, \ldots, X_T} f(x_{t+1}, \ldots, x_T \mid x_1, \ldots, x_t)
\]
Backtracking
Heuristic

\[ f(v^1_v, v^2_v, v^3_v) \rightarrow h(v^1_v, v^2_v, v^3_v) \]

- point model → true max total reward
- coverage model → estimated max total reward guarantee
Expanding tree
Anytime algorithm framework

- Exhaustive enumeration
  - depth-first recursive traverse
  - node $\iff$ subpath

- node in an expanding tree
- vertex in a multi-partite graph
Estimated reward $\leq$ Current best reward $\implies$ Stop exploring subpath

\[ f(path(n_3^{(1)})) + \hat{h}(n_3^{(1)}) \leq f(path(n_5^{(1)})) \]

\[ f(path(n_4^{(1)})) \leq f(path(n_4^{(1)})) \]

\[ f(path(n_4^{(2)})) + \hat{h}(n_4^{(2)}) \leq f(path(n_5^{(1)})) \]
Flow
Anytime algorithm framework

Start -> Recursive Backtracking -> Generate Solution

Tree Expanding <- Recursive Backtracking

Node Frozen <- Recursive Backtracking

Stop? (Yes/No) -> End

Solution
Lemma

Backtracking in Algorithm 1 never \textit{underestimates} the maximum total reward, which means

\[ \forall t \geq t', \hat{u}(x_t \mid v_1, \cdots, v_{t'}) \geq u(x_t \mid v_1, \cdots, v_{t'}). \]

Theorem

The anytime algorithm framework in Algorithm 4 can always find an \textit{optimal} solution given enough time.
A robot Wingman problem

Robot Wingman
Labelling
Robot wingman

Gazebo world

Map

Labeling
Path planning
Robot wingman

Path planning
Waypoints
Robot execution
**Problem size**
\[ \text{nodeNum(} \text{fully expanding tree} \text{)} \]

**Percentage of nodes explored**
\[ \frac{\text{nodeNum(} \text{current expanding tree} \text{)}}{\text{nodeNum(} \text{fully expanding tree} \text{)}} \]

**Percentage of optimal at first iteration**
\[ \frac{\text{score(} \text{first found solution} \text{)}}{\text{score(} \text{optimal solution} \text{)}} \]

**Number of iterations to reach optimal (normalized)**
\[ \frac{\text{iterationCount(} \text{optimal found} \text{)}}{\text{iterationCount(} \text{finish tree expanding} \text{)}} \]
Results

**Metrics**

**Quality of heuristic**
- Percentage of optimal at first iteration

**Quality of algorithm**
- Number of iterations to reach optimal (normalized)

![Graph showing metrics](image)
Performance

Results

average on the results of 20 runs @ random pattern

Problem size

Percentage of nodes explored

Percentage of optimal at first iteration

Number of iterations to reach optimal (normalized)
Information pattern difference
Robustness

Uniform
Random
Multi-Modal

Percentage of optimal at first iteration (%)

Percentage of nodes explored (%)

Percentage of optimal at first iteration
Human path difference

Robustness

- Line
- Spiral
- Lawn mower
- Arc
- Loitering
Human path difference

Robustness

Problem size

Percentage of nodes explored

Percentage of optimal at first iteration
Summary

- Search space reduction by **human constraint**
- Effectiveness and efficiency of **backtracking** on a multi-partite graph

Futurework

- Efficiency increase $\rightarrow$ Over-estimation reduction
- Offline planning $\rightarrow$ Online planning
- Single objective $\rightarrow$ Multiple objectives
Thank you!