Informative Path Planning with a Human Path Constraint

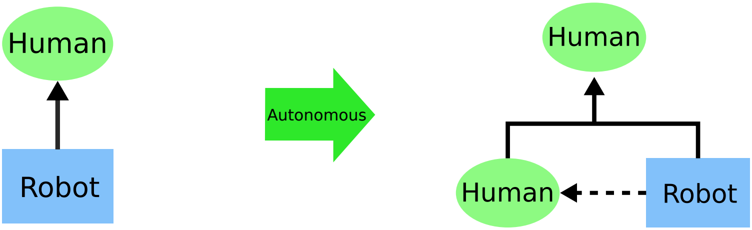
Daqing Yi

Department of Computer Science
Brigham Young University

Outline
Structure

Human-robot collaboration

Introduction

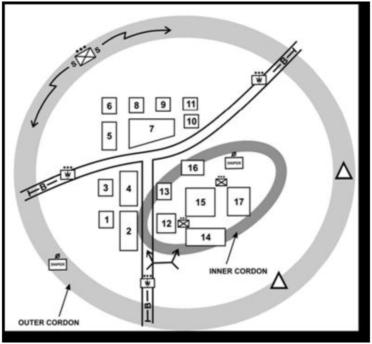


Human-robot interaction

Human-robot collaboration

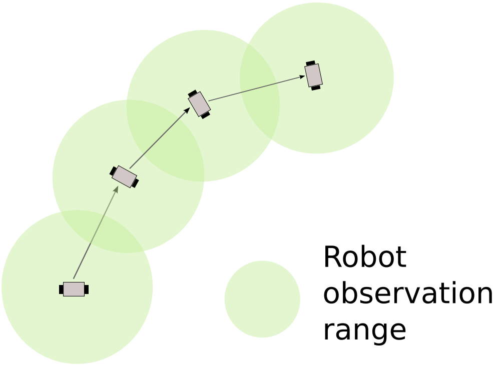
Cordon and search

Introduction



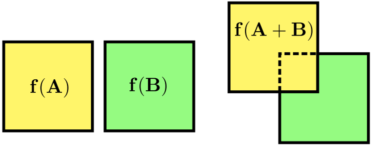
Coverage model
Informative path

- Maximum coverage problem
- Information gain - entropy



Submodularity

Informative path



Information

- search space S
- the observation of a robot $\mathbf{O}^X$
- the observation of a human $\mathbf{O}^Y$

$$f(A) + f(B) \geq f(A + B)$$

$$f(\mathbf{S}, \mathbf{O}^X) + f(\mathbf{S}, \mathbf{O}^{Y^h}) \geq f(\mathbf{S}, \mathbf{O}^X, \mathbf{O}^{Y^h})$$

Submodular orienteering

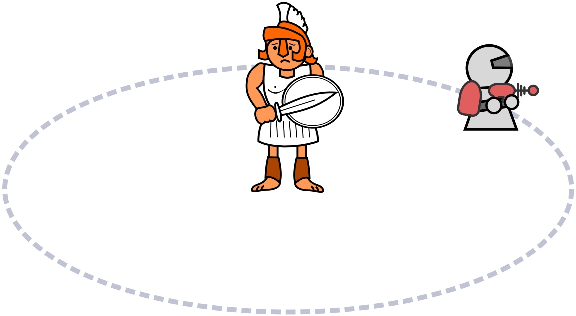
Informative path

Conditional mutual information

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

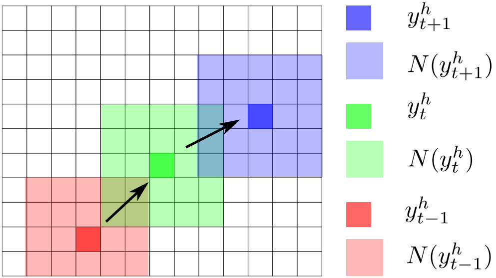
- Entropy reduction
- Submodularity
- Chain rule
  $I(\mathbf{S}; \mathbf{O}^X \mid \mathbf{O}^{Y^h}) = \sum_{t=1}^{T} I(O_t^X; \mathbf{S} \mid O_1^X, \cdots, O_{t-1}^X, \mathbf{O}^{Y^h})$

Team role
Human constraint



- cooperative observation
- assistance and protection

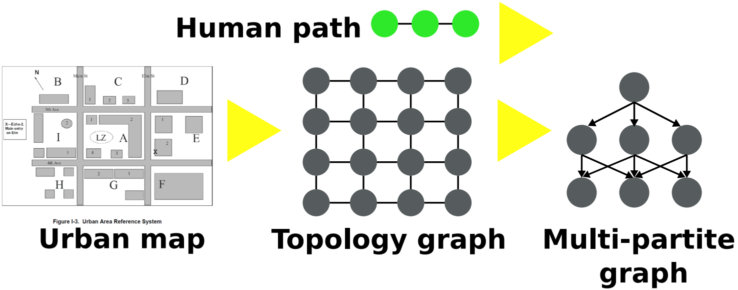
Neighboring function
Human constraint



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

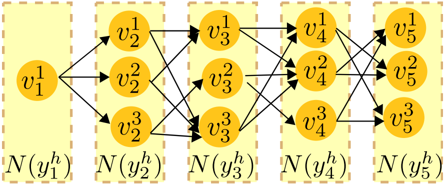
Problem abstraction

The optimization model



The multi-partite graph

The optimization model



A pruning process

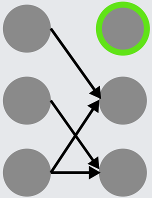
The optimization model

Forward pruning

Reachable

$\forall t \in \{2, \cdots T\},$

$\forall v \in V(t), deg^{-}(v) > 0$

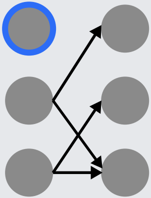


Backward pruning

Non-terminating

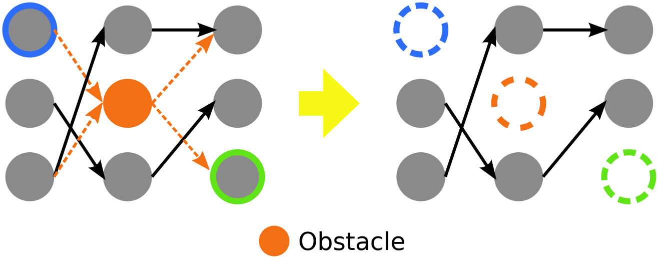
$\forall t \in \{1, \cdots T-1\},$

$\forall v \in V(t), deg^{+}(v) > 0$



Obstacles

The optimization model



Submodular orienteering on a multi-partite graph

The optimization problem

$$\begin{aligned}
\textit{Objective}: & X^* = \arg\max_{X} f(X); \\
\textit{Constraint}: & |X| = T, x_t \in V(t), (x_t, x_{t+1}) \in E.
\end{aligned}$$

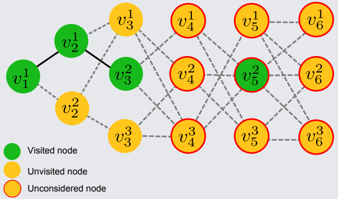
Bellman-like equation

Heuristic

$$\hat{x}_t = \arg\max_{X_t}\left[f(x_t \mid x_1, \cdots, x_{t-1}) + \max_{X_{t+1},\cdots,X_T} f(x_{t+1}, \cdots, x_T \mid x_1, \cdots, x_t)\right]$$

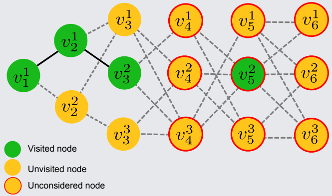
Maximum future reward

$$\max f(x_6, \cdots x_T \mid v_1^1, v_2^1, v_3^2, v_5^2)$$



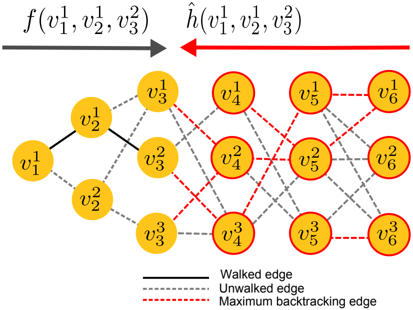
Maximum total reward

$$f(v_5^2 \mid v_1^1, v_2^1, v_3^2) \qquad h(v_1^1, v_2^1, v_3^2, v_5^2)$$



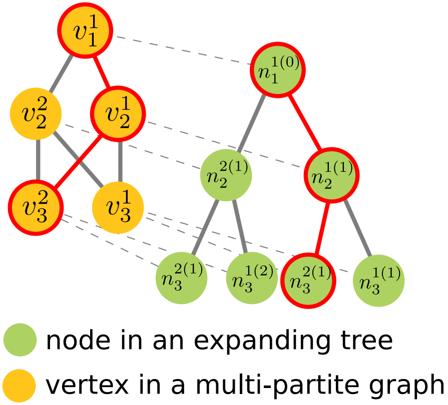
Backtracking

Heuristic



Expanding tree

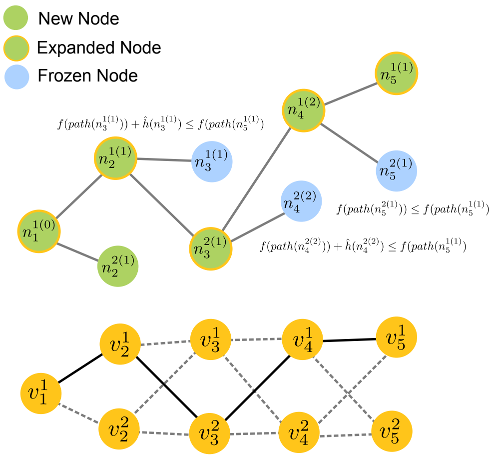
Anytime algorithm framework



- depth-first recursive traverse
- tracking the search process
- estimation storage

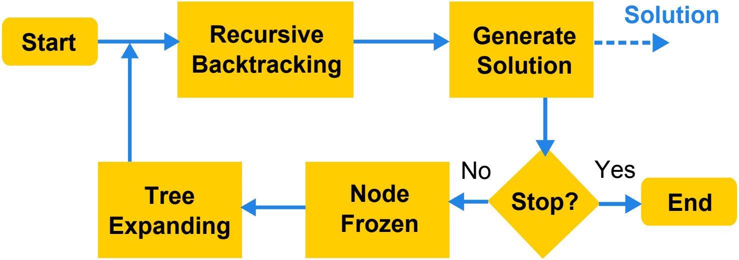
Node freeze

Anytime algorithm framework



Flow

Anytime algorithm framework



Performance guarantee

Anytime algorithm framework

Lemma

Backtracking in Algorithm 1 never underestimates the maximum total reward, which means

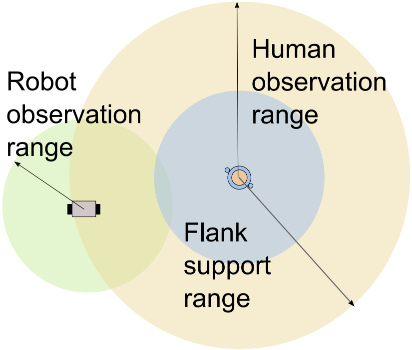
$$\forall t \geq t', \hat{u}(x_t \mid v_1, \cdots, v_{t'}) \geq u(x_t \mid v_1, \cdots, v_{t'}).$$

Theorem

The anytime algorithm framework in Algorithm 4 can always find an optimal solution given enough time.

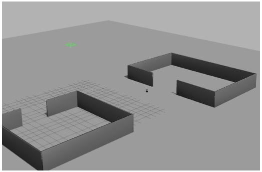
A robot Wingman problem

Robot Wingman

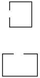


Labelling

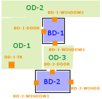
Robot wingman



World in Gazebo simulator



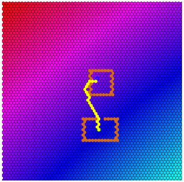
World map



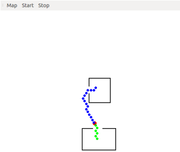
Labelled world map

Path planning

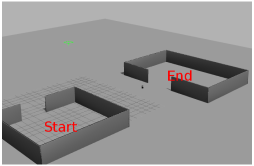
Robot wingman



Hexagonal map



Planned path



Robot execution

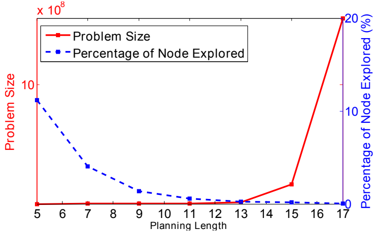
Metrics

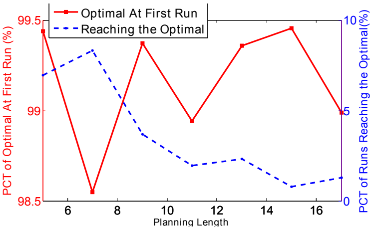
Results

- **Problem size**
  nodeNum(fully expanding tree)
- **Percentage of nodes explored**
  nodeNum(current expanding tree) / nodeNum(fully expanding tree)
- **Percentage of optimal at first run**
  score(first found solution) / score(optimal solution)
- **Percentage of runs reaching the optimal**
  iterationCount(optimal found) / iterationCount(finish tree expanding)

Performance

Results

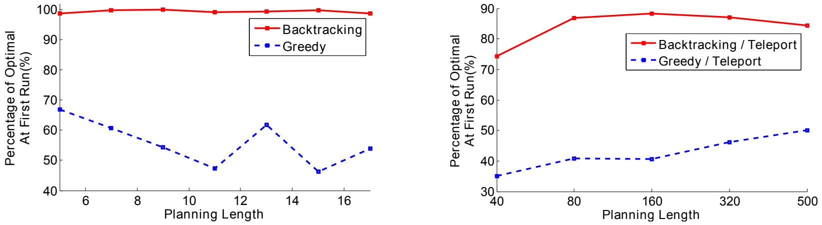




Compare with greedy heuristic

Performance

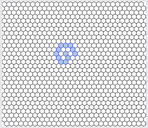
The performance of the heuristic (Percentage of optimal at first run)



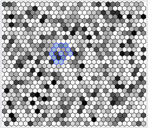
Environment difference

Robustness

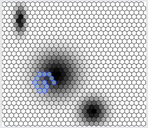
Uniform

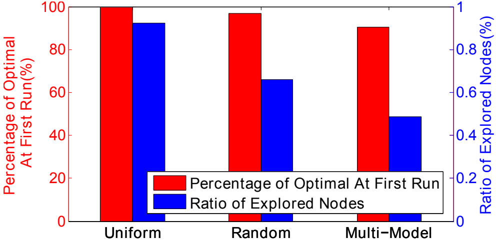


Random



Multi-Modal

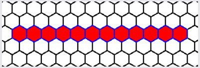




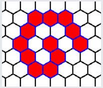
Human path difference

Robustness

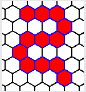
Line



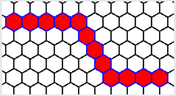
Spiral



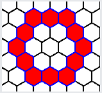
Lawn mower



Arc

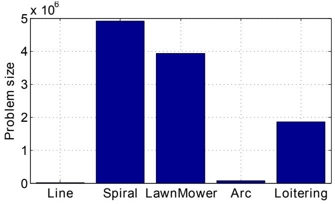


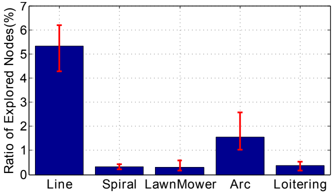
Loitering

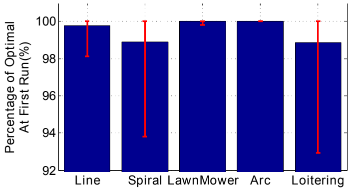


Human path difference

Robustness







Summary

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

- Vertex duplication in multi-partite graph $\rightarrow$ Over-estimation increase
- Offline planning $\rightarrow$ Online planning
- Single objective $\rightarrow$ Multiple objectives

Thank you!