Informative Path Planning with a Human Path Constraint

Daqing Yi Michael A. Goodrich Kevin D. Seppi

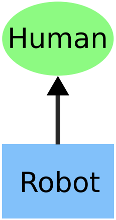
Department of Computer Science
Brigham Young University

Outline

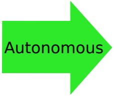
Structure

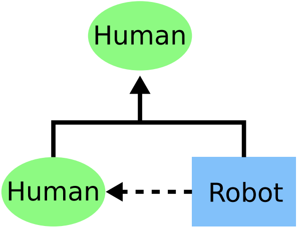
Human-robot collaboration

Introduction



Human-robot interaction

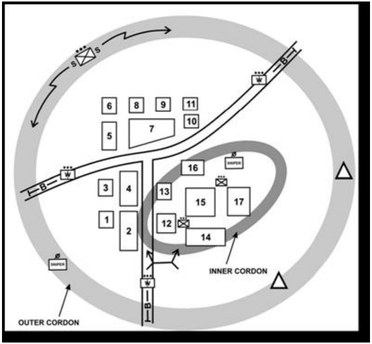




Human-robot collaboration

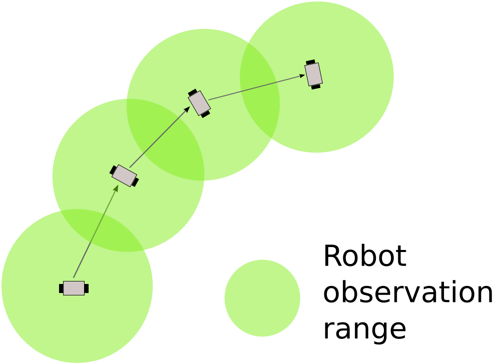
Cordon and search

Introduction



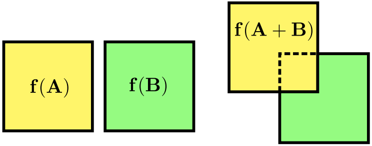
Coverage model
Informative path

- Information measurement - entropy
- Maximum coverage problem



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

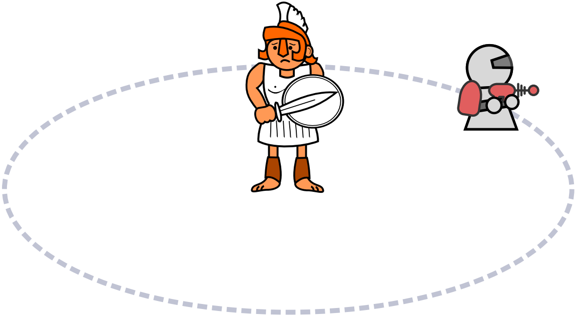
Conditional mutual information

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

- 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

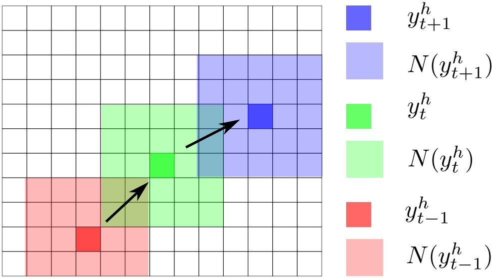
Human constraint



- cooperative observation
- assistance and protection

Neighboring function

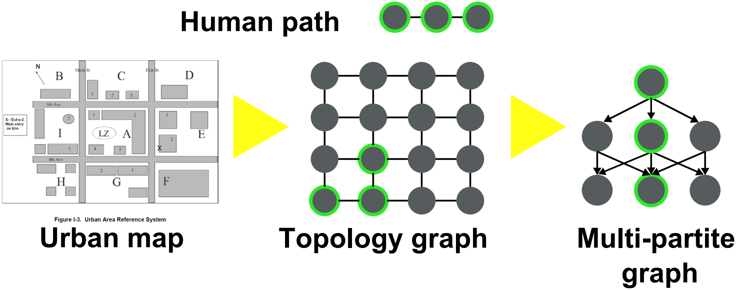
Human constraint



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

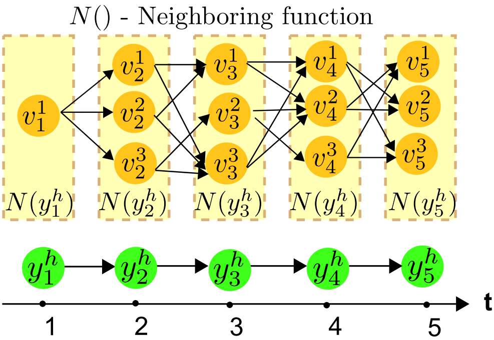
Problem abstraction

The optimization model



The multi-partite graph

The optimization model



- time-space synchronization
- connection determined by discretized map

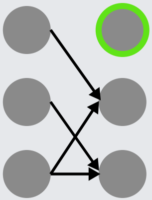
A pruning process

The optimization model

Reachable

Forward pruning

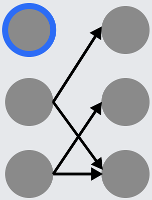
$\forall t \in \{2, \cdots T\}$,
$\forall v \in V(t), deg^{-}(v) > 0$



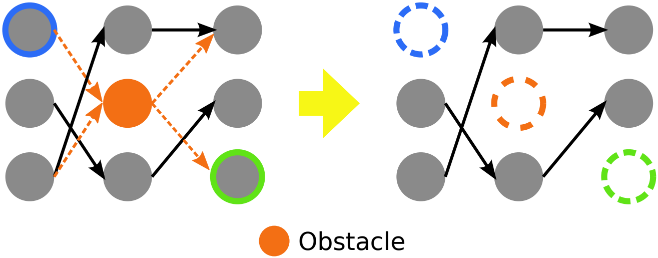
Non-terminating

Backward pruning

$\forall t \in \{1, \cdots T-1\}$,
$\forall v \in V(t), deg^{+}(v) > 0$



Obstacles
The optimization model



Submodular orienteering on a multi-partite graph

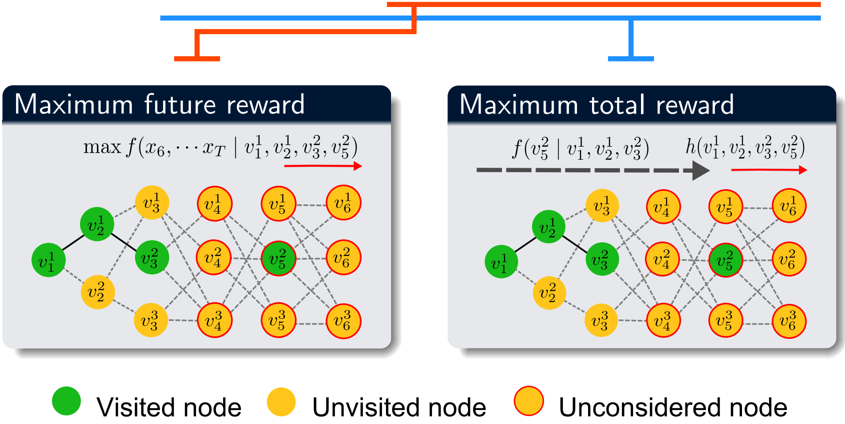
The optimization problem

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

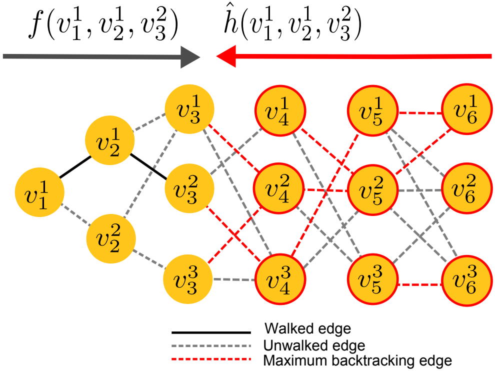
Bellman-like equation

Heuristic

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



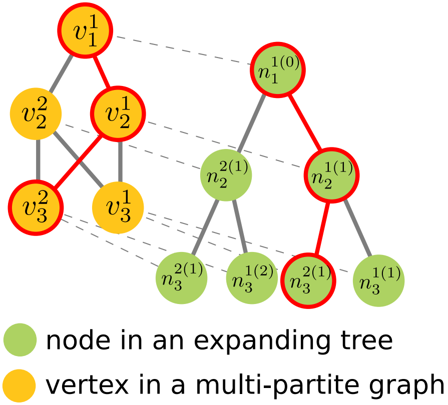
Backtracking
Heuristic



- point model $\rightarrow$ true max total reward
- coverage model $\rightarrow$ estimated max total reward guarantee

Expanding tree

Anytime algorithm framework



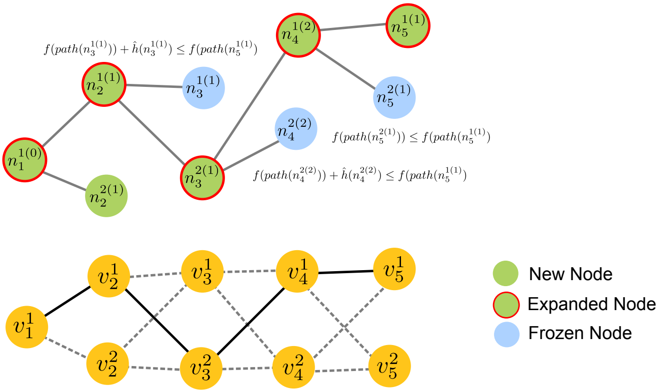
Exhaustive enumeration

- depth-first recursive traverse
- node $\Longleftrightarrow$ subpath

Node freeze

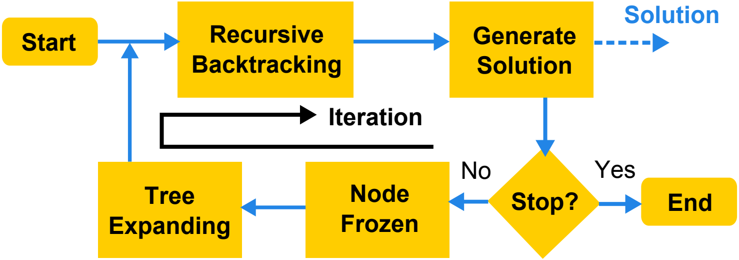
Anytime algorithm framework

Estimated reward $\leq$ Current best reward $\implies$ Stop exploring subpath



Flow

Anytime algorithm framework



Performance guarantee
Anytime algorithm framework

Lemma

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

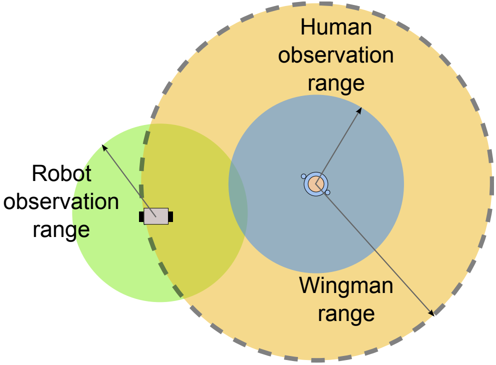
$$\forall t \geq t', \hat{u}(x_t \mid v_1, \cdots, v_{t'}) \geq u(x_t \mid v_1, \cdots, v_{t'}).$$

Theorem

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

A robot Wingman problem

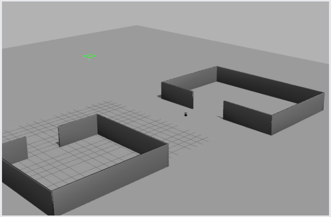
Robot Wingman



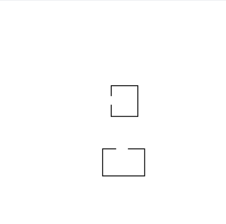
Labelling

Robot wingman

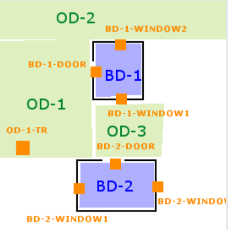
Gazebo world



Map

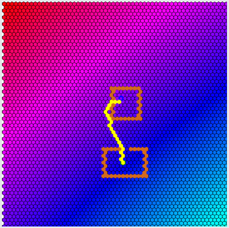


Labeling

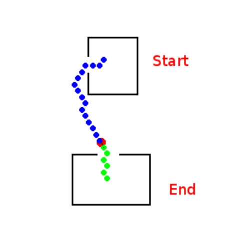


Path planning
Robot wingman

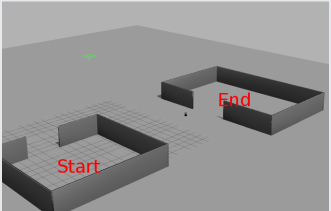
Path planning



Waypoints



Robot execution



Metrics
Results

- **Problem size**
  nodeNum(fully_expanding_tree)

- **Percentage of nodes explored**
  nodeNum(current_expanding_tree) / nodeNum(fully_expanding_tree)

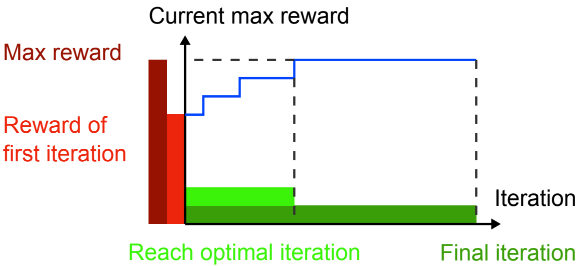
- **Percentage of optimal at first iteration**
  score(first_found_solution) / score(optimal_solution)

- **Number of iterations to reach optimal (normalized)**
  iterationCount(optimal_found) / iterationCount(finish_tree_expanding)

Metrics
Results

quality of heuristic

Percentage of optimal at first iteration



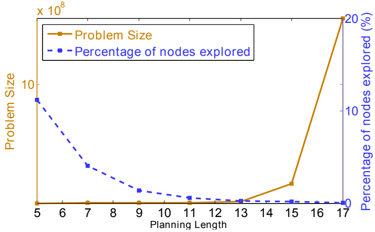
quality of algorithm

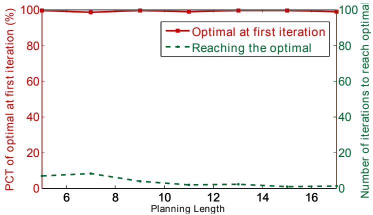
Number of iterations to reach optimal (normalized)

Performance

Results

average on the results of 20 runs @ random pattern





Problem size

Percentage of nodes explored

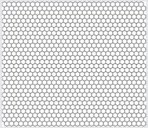
Percentage of optimal at first iteration

Number of iterations to reach optimal (normalized)

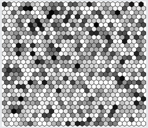
Information pattern difference

Robustness

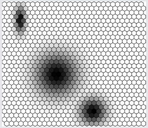
Uniform

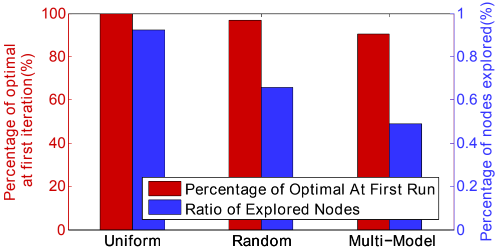


Random



Multi-Modal





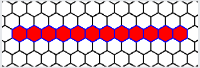
Percentage of optimal at first iteration

Percentage of nodes explored

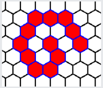
Human path difference

Robustness

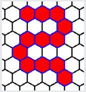
Line



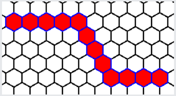
Spiral



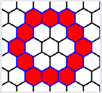
Lawn mower



Arc

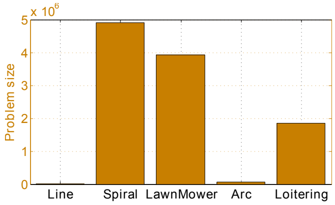


Loitering

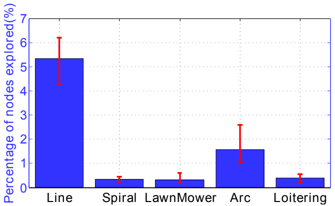


Human path difference

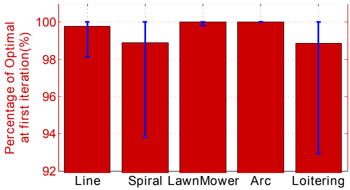
Robustness



Problem size



Percentage of nodes explored



Percentage of optimal at first iteration

Summary and futurework

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!