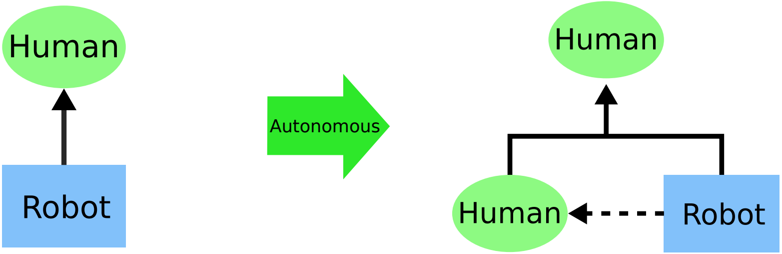
Informative Path Planning with a Human Path Constraint

Daqing Yi

Department of Computer Science
Brigham Young University

June 24, 2014

Outline

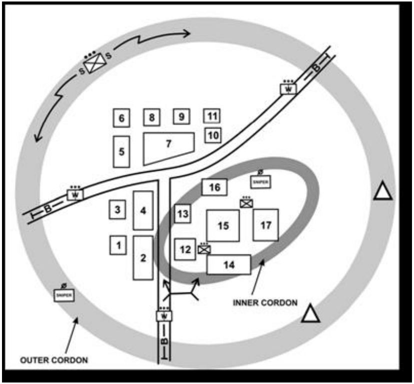


Human-robot interaction

Human-robot collaboration

Cordon and search

Introduction

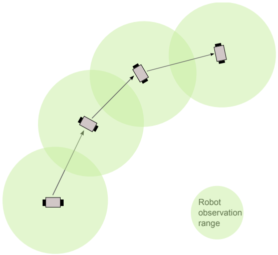


Outline

Coverage model

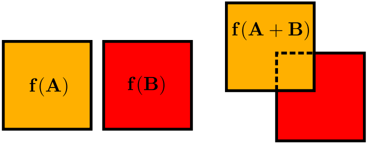
Informative path

- Maximum coverage problem
- Entropy-based set function



Submodularity

Informative path



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

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

Conditional mutual information

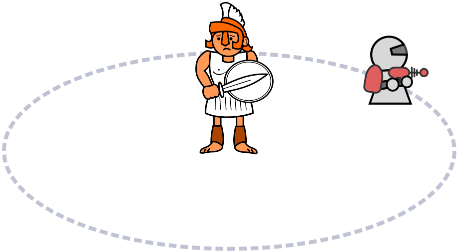
$I(\mathbf{S}; \mathbf{O}^X \mid \mathbf{O}^{Y^h}) = H(\mathbf{S} \mid \mathbf{O}^{Y^h}) - H(\mathbf{S} \mid \mathbf{O}^X, \mathbf{O}^{Y^h})$

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

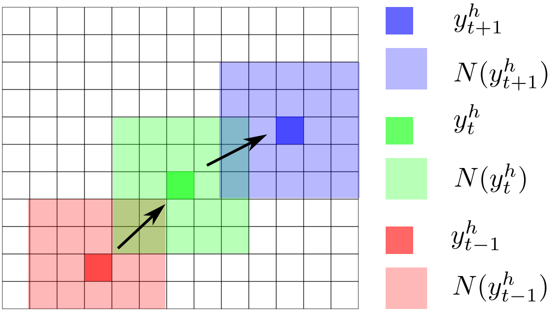
Outline

Team role

Human constraint



- cooperative observation
- assistance and protection

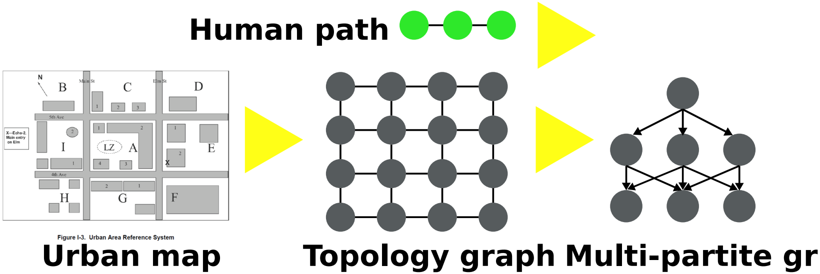


- human path $\{y_1^h \cdots y_T^h\}$
- neighboring function $N(y_t^h)$

Outline

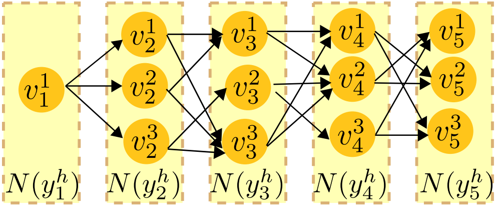
Problem abstraction

The optimization model



The multi-partite graph

The optimization model



Multi-partite graph
$G = (V, E, T)$

- T - partition number
- $V = \cup_{t=1}^{T} V(t)$
- $(v_t^i, v_{t+1}^j) \in E$

A pruning process

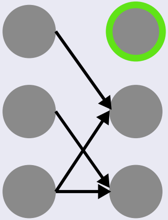
The optimization model

Forward pruning

Reachable

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

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

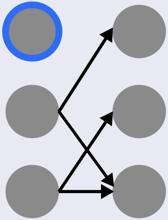


Backward pruning

Non-terminating

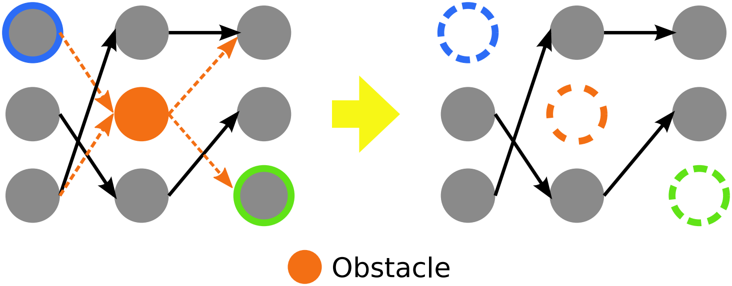
$\forall t \in \{1, \cdots T-1\}$,

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



Obstacles

The optimization model



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

Outline

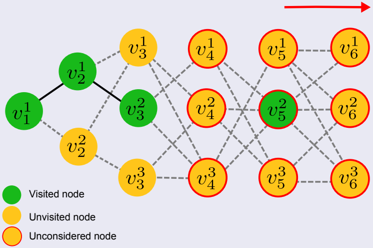
Bellmen-like equation

Heuristic

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

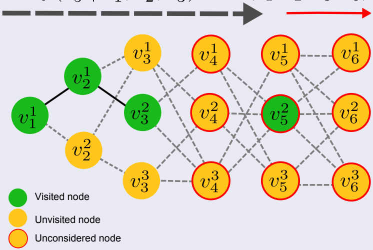
Maximum future reward

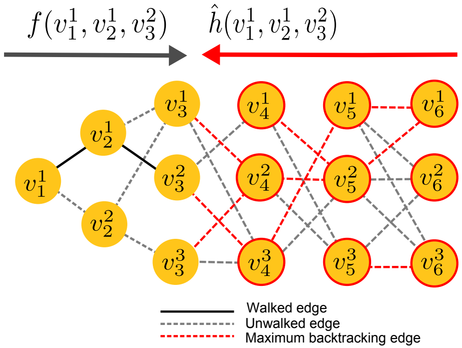
$$\max f(x_6, \cdots x_T \mid v_1^1, v_2^1, v_3^2, v_5^2)$$



Maximum total reward

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



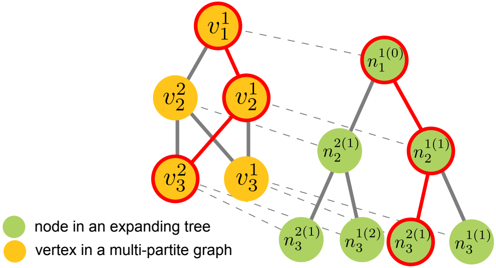


$f(v_1^1, v_2^1, v_3^2)$
$\hat{h}(v_1^1, v_2^1, v_3^2)$
v_1^1
v_2^1
v_2^2
v_3^1
v_3^2
v_3^3
v_4^1
v_4^2
v_4^3
v_5^1
v_5^2
v_5^3
v_6^1
v_6^2
v_6^3
Walked edge
Unwalked edge
Maximum backtracking edge

Outline

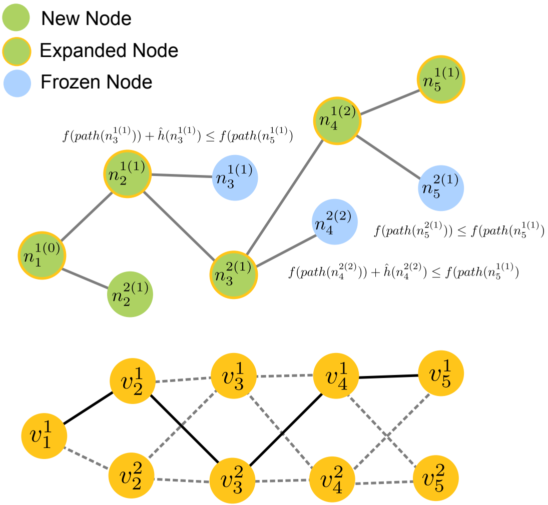
Expanding tree

Anytime algorithm framework



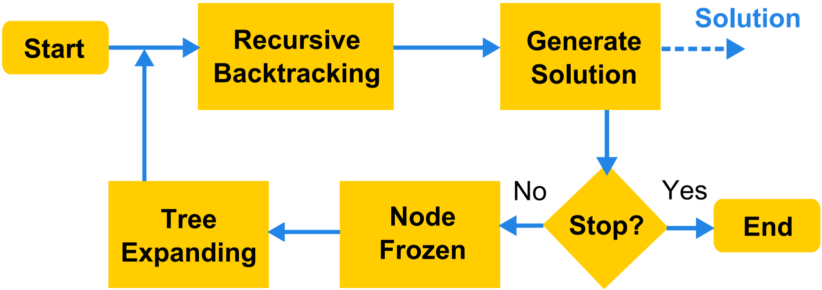
Node freeze

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

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

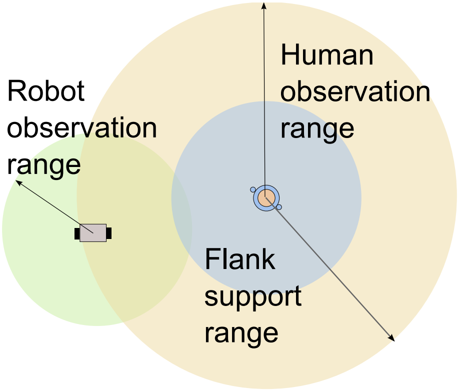
Theorem

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

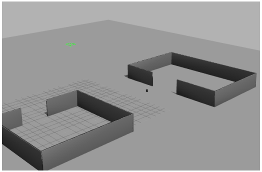
Outline

A robot Wingman problem

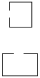
Robot Wingman



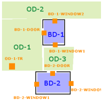
Labelling
Robot wingman



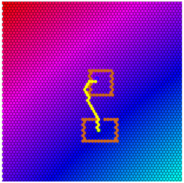
World in Gazebo simulator



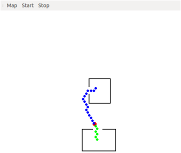
World map



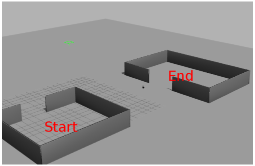
Labelled world map



Hexagonal map



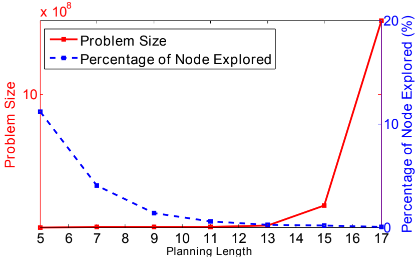
Planned path

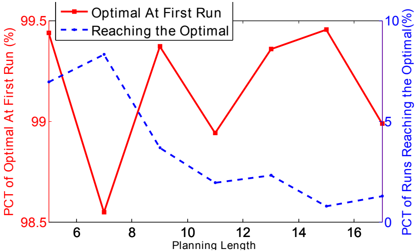


Robot execution

Outline

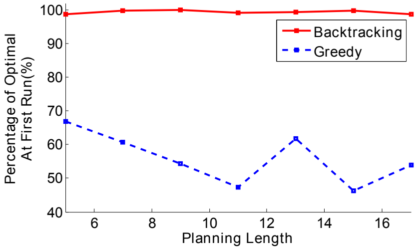
- **Problem size**
  nodeNum(fully expanding tree)
- **percentage of nodes explored**
  nodeNum(current expanding tree) / nodeNum(fully expanding tree)
- **Percentage of optimal at first run**
  score(first found solution) / score(optimal solution)
- **Percentage of runs reaching the optimal**
  iterationCount(optimal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

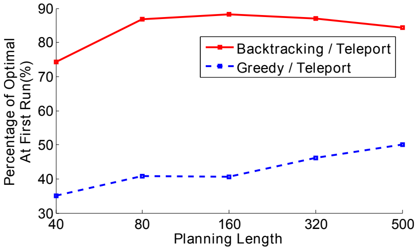

x 10^8
Problem Size
Percentage of Node Explored
Problem Size
10
Percentage of Node Explored (%)
20
10
0
5 6 7 8 9 10 11 12 13 14 15 16 17
Planning Length


Optimal At First Run
Reaching the Optimal
PCT of Optimal At First Run (%)
99.5
99
98.5
PCT of Runs Reaching the Optimal(%)
10
5
0
6 8 10 12 14 16
Planning Length

Compare with greedy heuristic
Performance

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

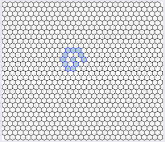




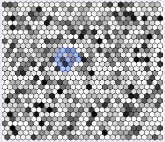
Environment difference

Robustness

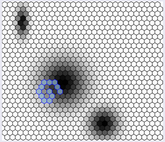
Uniform

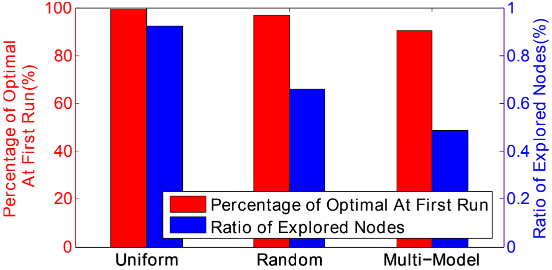


Random



Multi-Modal

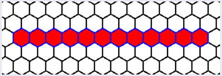




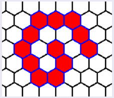
Human path difference

Robustness

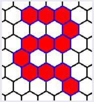
Line



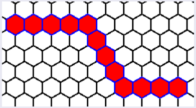
Spiral



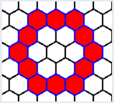
Lawn mower



Arc

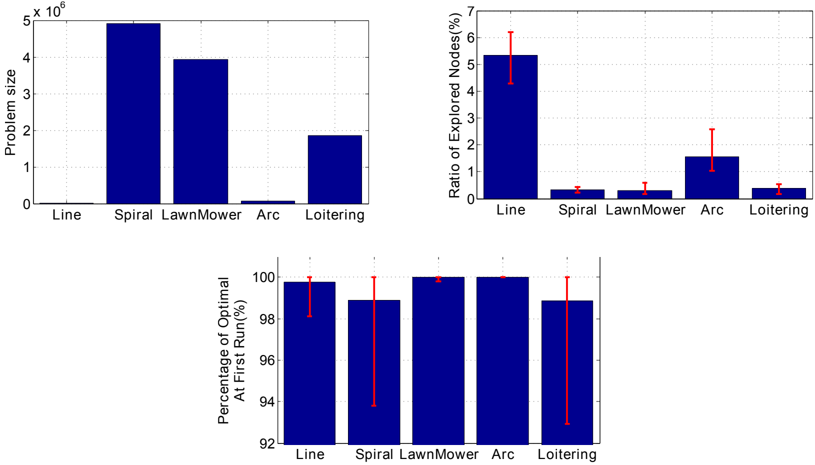


Loitering



Human path difference

Robustness



Summary

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

- Vertices duplication in multi-partite graph → Over-estimation increase
- Offline planning → Online planning
- Single objective → Multiple objectives

Thank you!