Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths

Tal Regev and Vadim Indelman

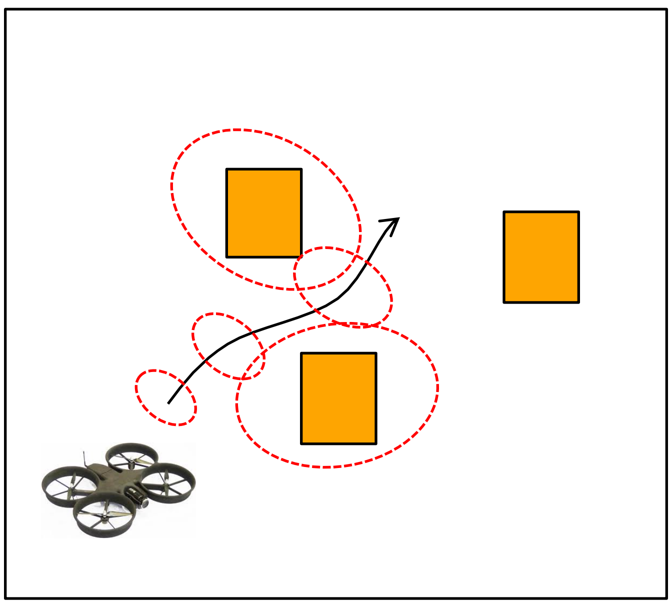
TECHNION
Israel Institute of Technology

International Conference on Intelligent Robots and Systems (IROS), October 2016

Introduction

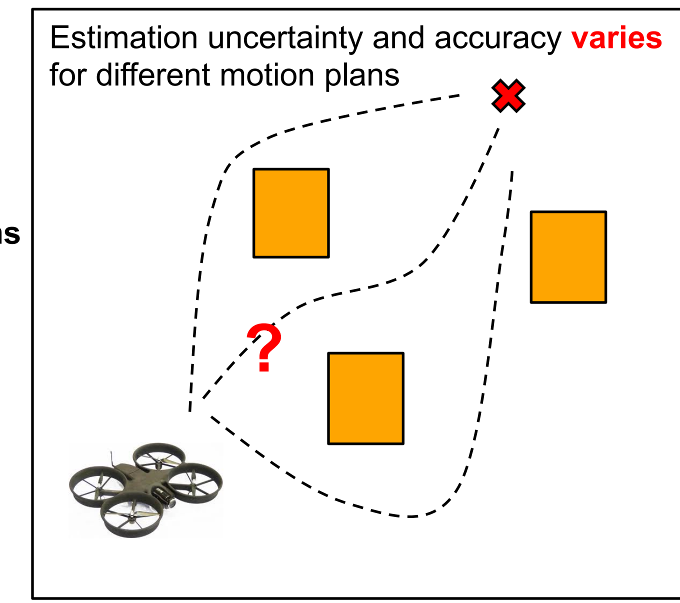
- Key components for autonomous operation include
  - **Perception**: Where am I? What is the surrounding environment?
  - **Planning**: What to do next?

Localization and mapping given robot motion



Coupled problems

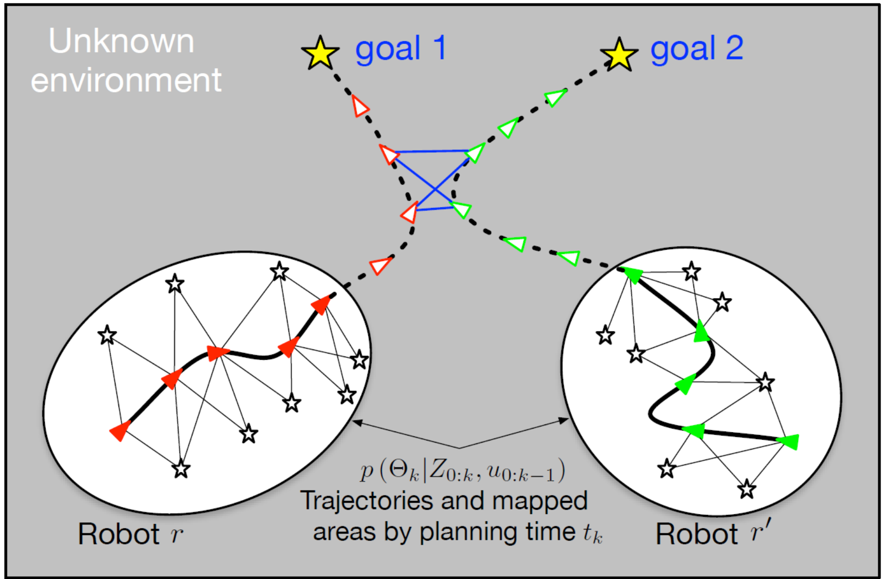
Planning (e.g. reach a goal)



Related Work – Belief Space Planning (BSP)

- **Belief space planning** (BSP) - fundamental problem in robotics
- Multi-robot framework has numerous advantages, e.g. higher accuracy

- Existing approaches typically assume **environment/map is known** [Prentice and Roy '09], [Miller et al. '09], [Platt et al. '10], [Van den Berg et al. '12], [Hollinger et al. '13]
- Recent research relaxes this assumption, **incorporates map uncertainty within the belief** [Valencia et al. '12], [Kim and Eustice '14], [Chaves et al. '14], [Indelman et al. '15a]
- Extension to **multi-robot centralized** setting [Indelman '15b] [Indelman '15c]
  - Reason about future observations of unknown scenes within multi-robot BSP

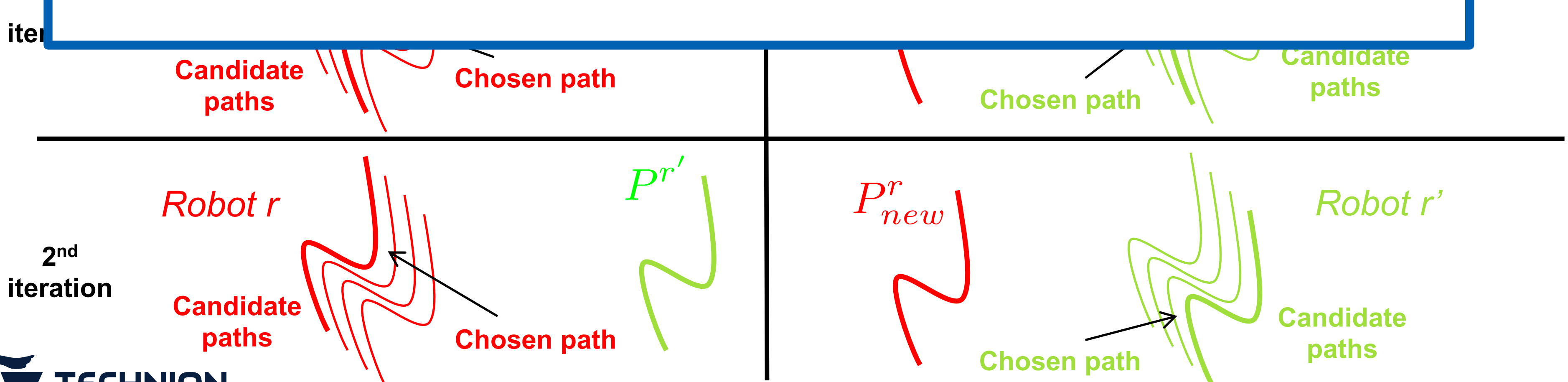


Related Work – Belief Space Planning (BSP)

- Discrete action space – each robot has a set of **candidate** paths/actions
- Optimal solution is intractable: involves all combinations of paths of all robots

- A common iterative (suboptimal) approach to reduce computational effort: [Atanasov '14 TRO] [Levine '13 JAIS]
  - Each robot **calculates** the **best solution and announces it** to other robots, given previous announced paths

Calculations are performed from scratch at each iteration

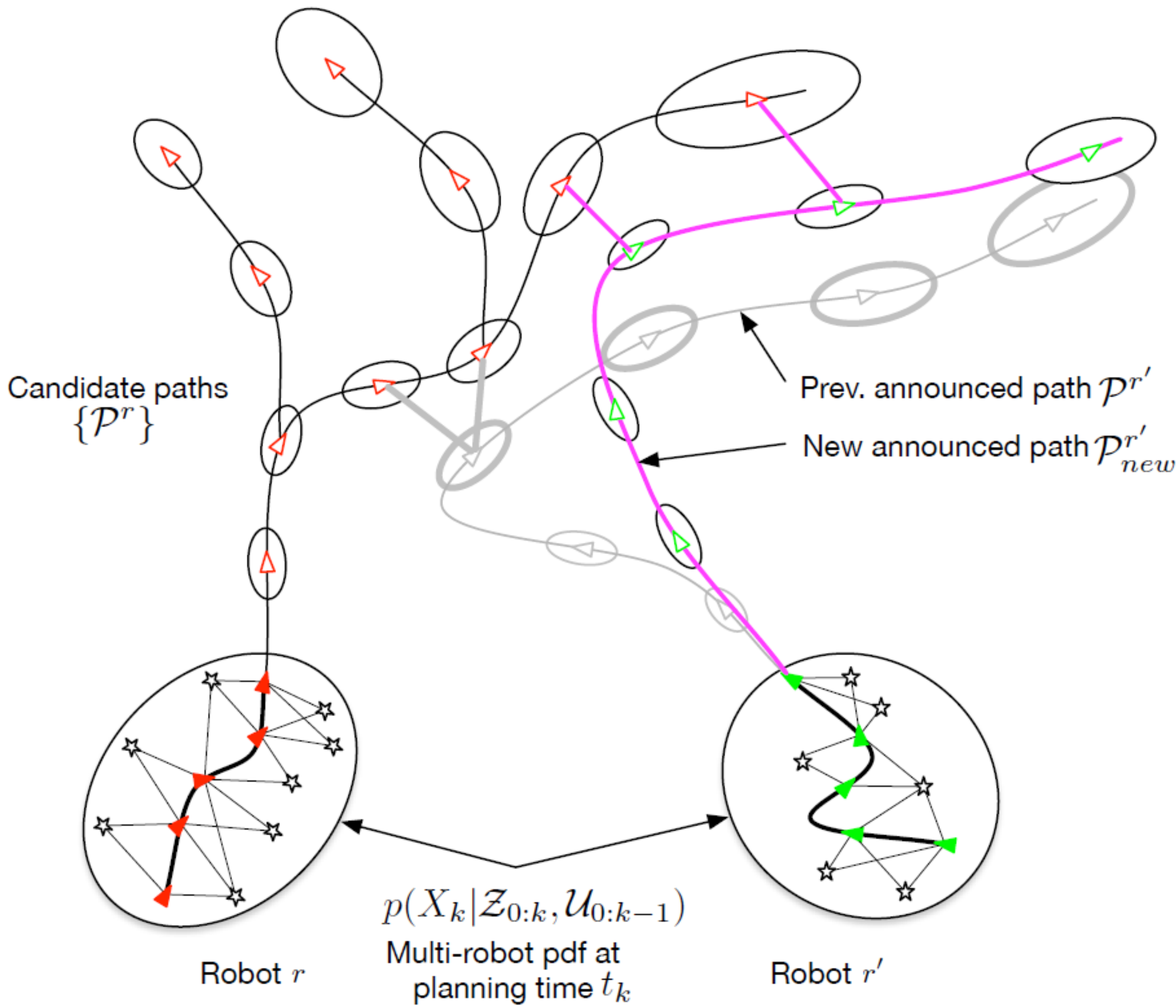


TECHNION Israel Institute of Technology

Contribution

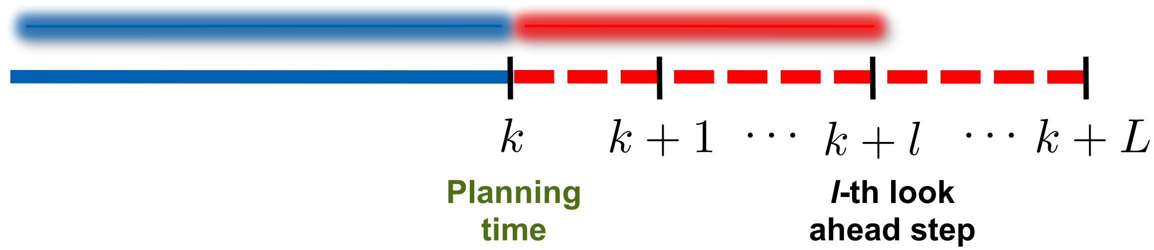
- Each time the announced path from some robot r' changes, each robot r has to re-evaluate its candidate paths

- **Key idea**
  - Not all paths are impacted due to change in the announced paths
  - Impacted paths can be efficiently re-evaluated by reusing calculations



Formulation - Multi-robot Belief Space Planning

- Belief for robot r at a future time t_{k+l}: $b\left[X^r_{k+l}\right] \doteq p\left(X^r_{k+l} | Z^r_{0:k+l}, u^r_{0:k+l-1}\right)$



- Multi-robot objective function:

$$J\left(u\right) = \mathbb{E}\left[\sum_{l=1}^{L}\sum_{r=1}^{R} c^r_l\left(b\left[X^r_{k+l}\right], u^r_{k+l}\right)\right]$$

- Optimal controls for all R robots:

$$u^{\star} = \arg\min_{u} J\left(u\right)$$

T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

Formulation - Multi-robot Belief Space Planning

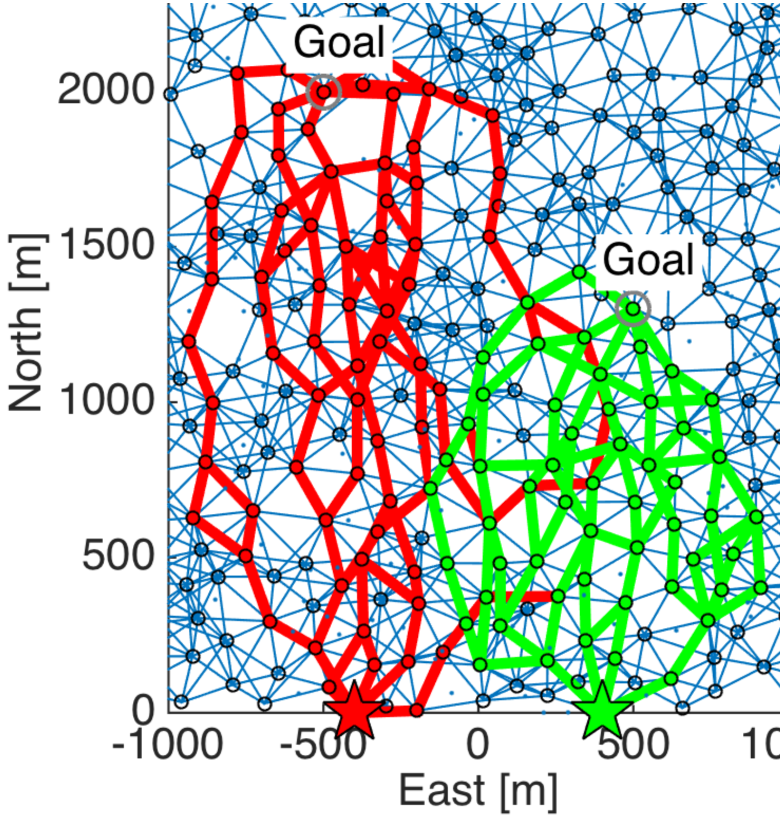
- Multi-robot probability distribution function (pdf) at planning time t_k :

$$b[\mathrm{P}] = p(X_k \mid \mathrm{Z}_{0:k}, \mathrm{U}_{0:k-1}) \prod_{r=1}^{R} \left[\prod_{l=1}^{L(\mathrm{P}^r)} p(x_{v_l}^r \mid x_{v_{l-1}}^r, u_{v_{l-1}}^r) \cdot p(Z_{v_l}^r \mid X_{k+l}^r) \prod_{\{i,j\}} p(z_{i,j}^{r,r'} \mid x_{v_i}^r, x_{v_j}^{r'}) \right]$$

- P - Some specific candidate paths for all robots $P \doteq \left\{ P^r, P^{r'}, \ldots \right\}$

- Maximum a posteriori (MAP) inference:

$$b[\mathrm{P}] = N\left(\hat{X}(\mathrm{P}), \Lambda^{-1}(\mathrm{P})\right)$$



Formulation - Multi-robot Belief Space Planning

- Multi-robot probability distribution function (pdf) at planning time t_k:

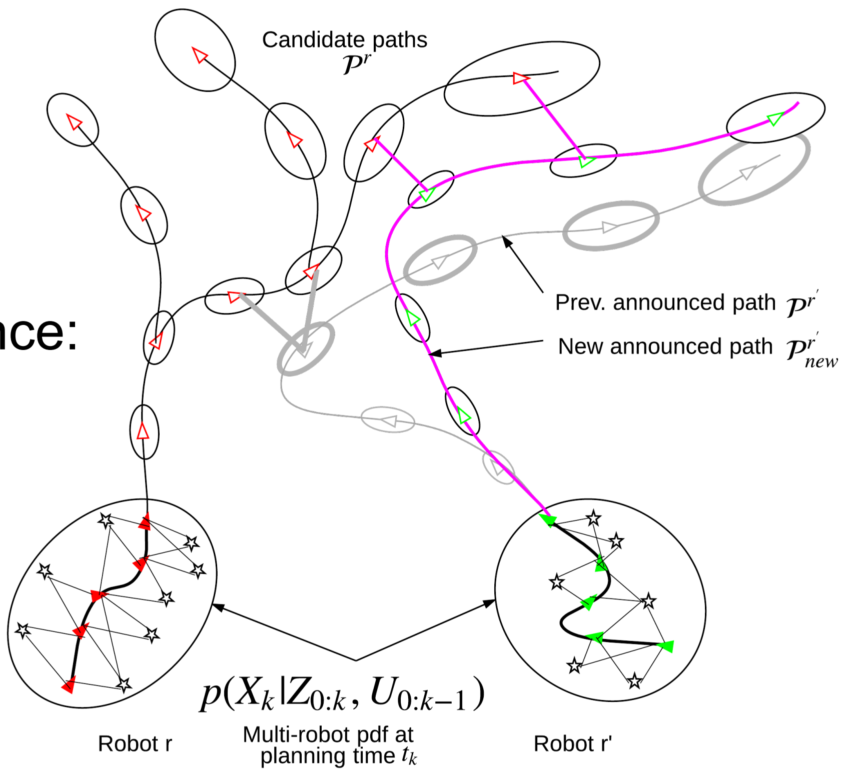
$$b[\mathrm{P}] = p(X_k \mid \mathrm{Z}_{0:k}, \mathrm{U}_{0:k-1}) \prod_{r=1}^{R} \left[\prod_{l=1}^{L(\mathrm{P}^r)} p(x^r_{v_l} \mid x^r_{v_{l-1}}, u^r_{v_{l-1}}) \cdot p(Z^r_{v_l} \mid X^r_{k+l}) \prod_{\{i,j\}} p(z^{r,r'}_{i,j} \mid x^r_{v_i}, x^{r'}_{v_j}) \right]$$

Prior

$$\Lambda(\mathrm{P}) = \Lambda_k + \sum_{r=1}^{R} \left[\sum_{l=1}^{L(\mathrm{P}^r)} \Lambda_l^{r,local} + \sum_{\{i,j\}} \Lambda^{r,r'}_{i,j} \right]$$

- Maximum a posteriori (MAP) inference:

$$b[\mathrm{P}] = N\left(\hat{X}(\mathrm{P}), \Lambda^{-1}(\mathrm{P})\right)$$



T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

Formulation - Multi-robot Belief Space Planning

- Multi-robot probability distribution function (pdf) at planning time t_k:

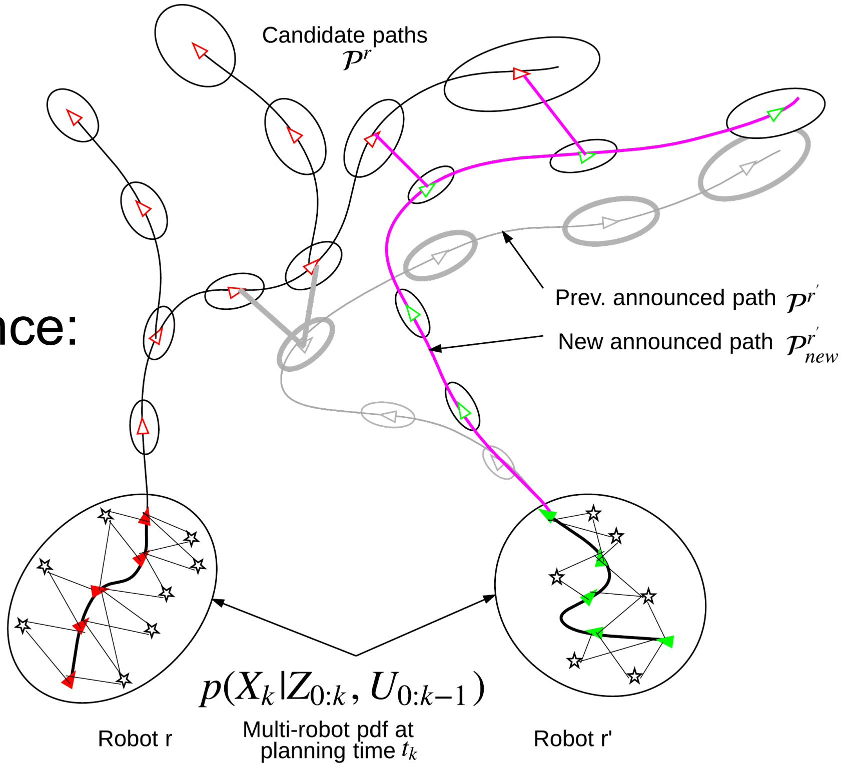
$$b[\mathrm{P}] = p(X_k \mid Z_{0:k}, U_{0:k-1}) \prod_{r=1}^{R} \left[\prod_{l=1}^{L(\mathrm{P}^r)} p(x^r_{v_l} \mid x^r_{v_{l-1}}, u^r_{v_{l-1}}) \cdot p(Z^r_{v_l} \mid X^r_{k+l}) \prod_{\{i,j\}} p(z^{r,r'}_{i,j} \mid x^r_{v_i}, x^{r'}_{v_j}) \right]$$

Prior   Local information

$$\Lambda(\mathrm{P}) = \Lambda_k + \sum_{r=1}^{R} \left[\sum_{l=1}^{L(\mathrm{P}^r)} \Lambda_l^{r,local} + \sum_{\{i,j\}} \Lambda^{r,r'}_{i,j} \right]$$

- Maximum a posteriori (MAP) inference:

$$b[\mathrm{P}] = N\left(\hat{X}(\mathrm{P}), \Lambda^{-1}(\mathrm{P})\right)$$



T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

Formulation - Multi-robot Belief Space Planning

- Multi-robot probability distribution function (pdf) at planning time t_k:

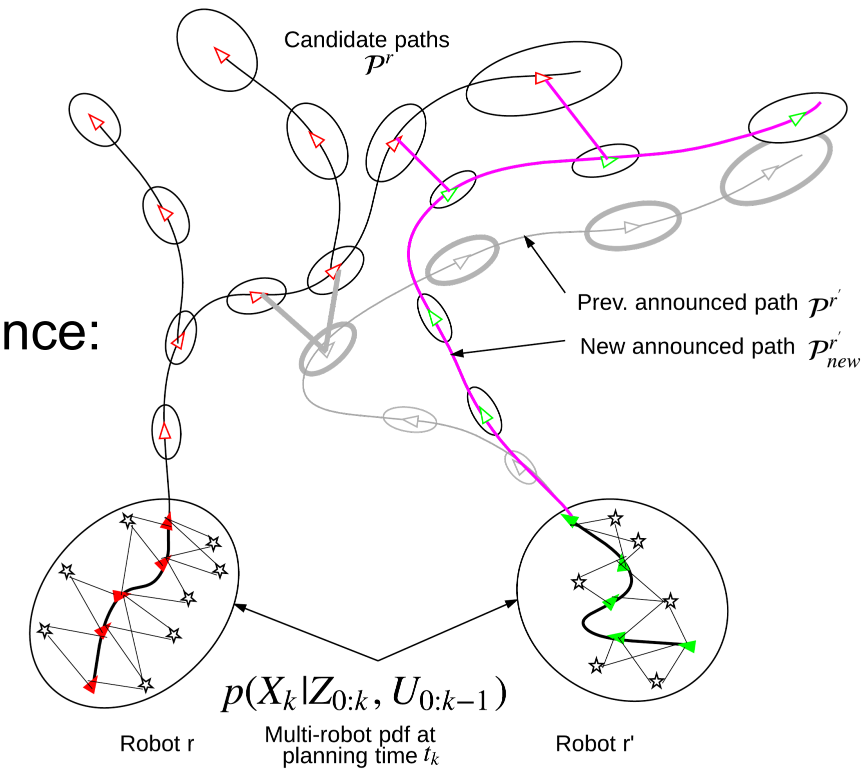
$$b[\mathrm{P}] = p(X_k \mid \mathrm{Z}_{0:k}, \mathrm{U}_{0:k-1}) \prod_{r=1}^{R} \left[\prod_{l=1}^{L(\mathrm{P}^r)} p(x^r_{v_l} \mid x^r_{v_{l-1}}, u^r_{v_{l-1}}) \cdot p(Z^r_{v_l} \mid X^r_{k+l}) \prod_{\{i,j\}} p(z^{r,r'}_{i,j} \mid x^r_{v_i}, x^{r'}_{v_j}) \right]$$

Prior Local information Mutual observations

$$\Lambda(\mathrm{P}) = \Lambda_k + \sum_{r=1}^{R} \left[\sum_{l=1}^{L(\mathrm{P}^r)} \Lambda_l^{r,local} + \sum_{\{i,j\}} \Lambda^{r,r'}_{i,j} \right]$$

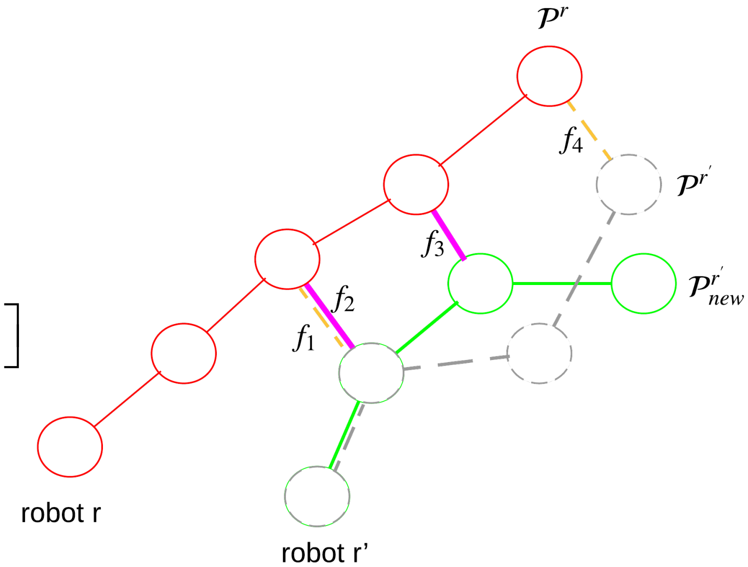
- Maximum a posteriori (MAP) inference:

$$b[\mathrm{P}] = N\left(\hat{X}(\mathrm{P}), \Lambda^{-1}(\mathrm{P})\right)$$



T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

Factor Graph Inference



- Joint state $\mathrm{P}^r, \mathrm{P}^{r'}$:

$$b[\mathrm{P}^r, \mathrm{P}^{r'}] = p(X_k \mid \mathrm{Z}_{0:k}, \mathrm{U}_{0:k-1}) \prod_{l=1}^{L(\mathrm{P}^r)} \left[p(x^r_{v_l} \mid x^r_{v_{l-1}}, u^r_{v_{l-1}}) \cdot p(Z^r_{v_l} \mid X^r_{k+l}) \right]$$

$$\prod_{l=1}^{L(\mathrm{P}^{r'})} \left[p(x^{r'}_{v_l} \mid x^{r'}_{v_{l-1}}, u^{r'}_{v_{l-1}}) \cdot p(Z^{r'}_{v_l} \mid X^{r'}_{k+l}) \right] \prod_{r=1}^{R} \left[\prod_{\{i,j\}} p(z^{r,r'}_{i,j} \mid x^r_{v_i}, x^{r'}_{v_j}) \right]$$

- Joint state $\mathrm{P}^r, \mathrm{P}^{r'}_{new}$:

$$b[\mathrm{P}^r, \mathrm{P}^{r'}_{new}] = p(X_k \mid \mathrm{Z}_{0:k}, \mathrm{U}_{0:k-1}) \prod_{l=1}^{L(\mathrm{P}^r)} \left[p(x^r_{v_l} \mid x^r_{v_{l-1}}, u^r_{v_{l-1}}) \cdot p(Z^r_{v_l} \mid X^r_{k+l}) \right]$$

$$\prod_{l=1}^{L(\mathrm{P}^{r'})} \left[p(x^{r'}_{v_l} \mid x^{r'}_{v_{l-1}}, u^{r'}_{v_{l-1}}) \cdot p(Z^{r'}_{v_l} \mid X^{r'}_{k+l}) \right] \prod_{r=1}^{R} \left[\prod_{\{i,j\}} p(z^{r,r'}_{i,j} \mid x^r_{v_i}, x^{r'}_{v_j}) \right]$$

Does not change

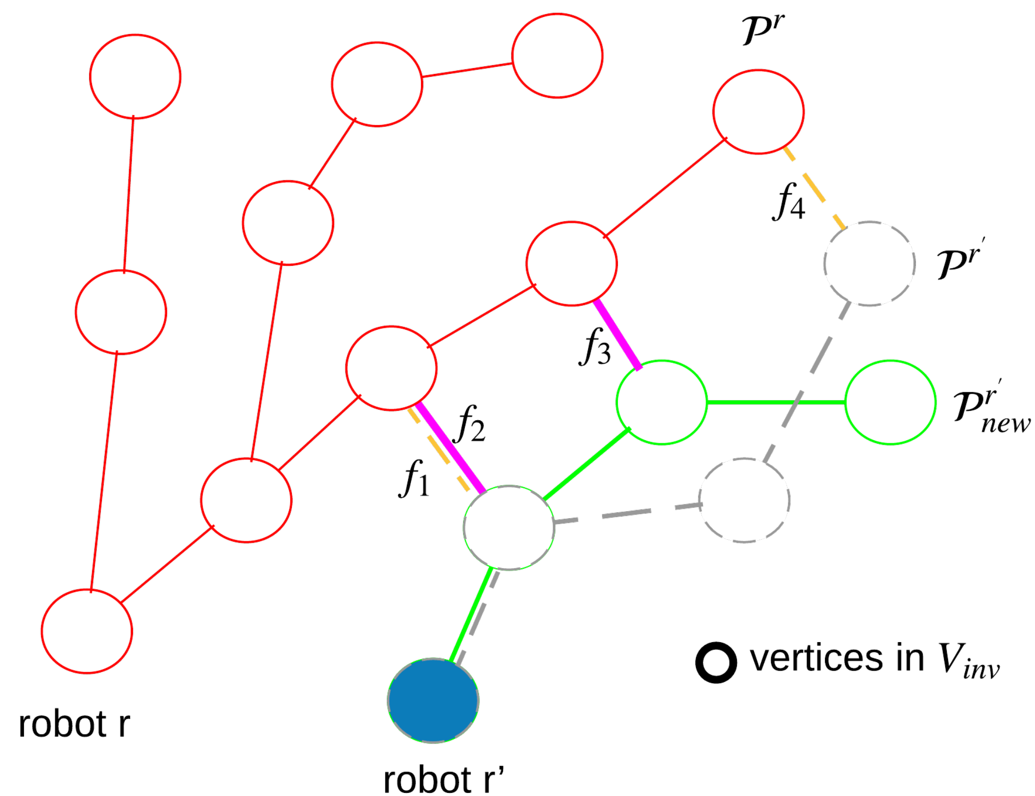
Algorithm Overview

- Recall **key idea**:
  - Not all paths are impacted due to change in the announced paths
  - Impacted paths can be efficiently re-evaluated by reusing calculations

- Iterate **over vertices** in previous and new announced paths
- Identify **involved** vertices in multi-robot factors – collect into set V_{inv}
- Identify and **mark** involved paths from all candidate paths

Algorithm Overview – Details

- Iterate **over vertices** in previous and new announced paths
- Identify **involved** vertices in multi-robot factors – collect into set V_{inv}
- Identify and **mark** involved paths from all candidate paths

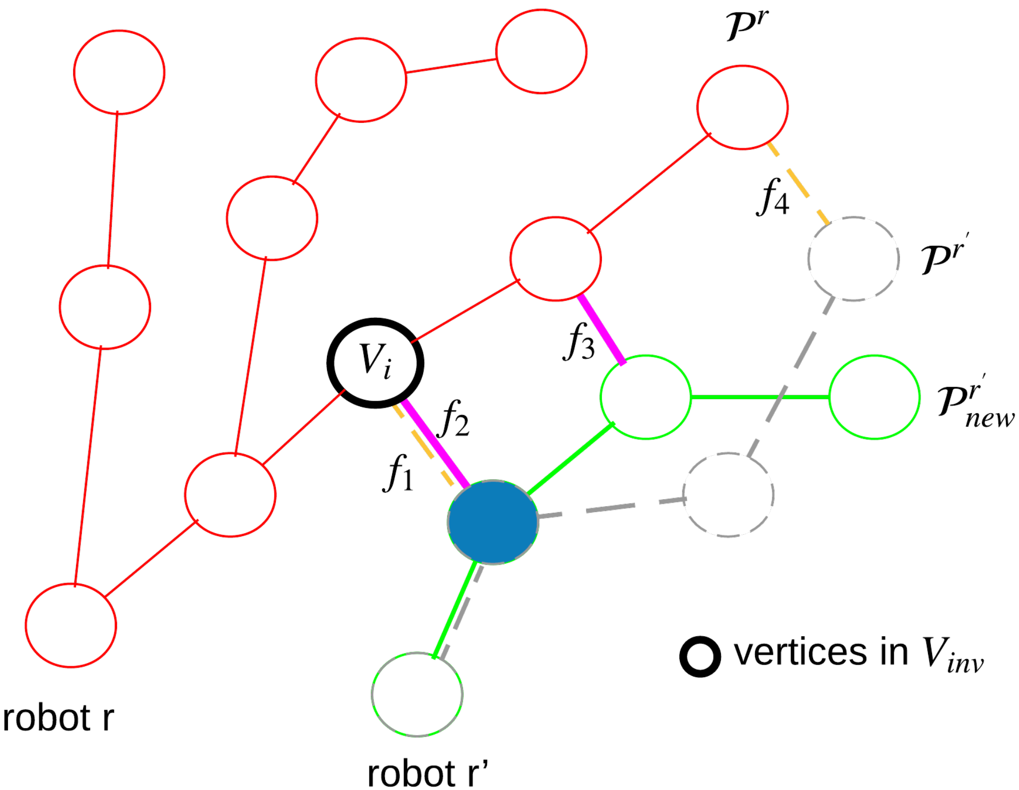


T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

TECHNION
Israel Institute of Technology

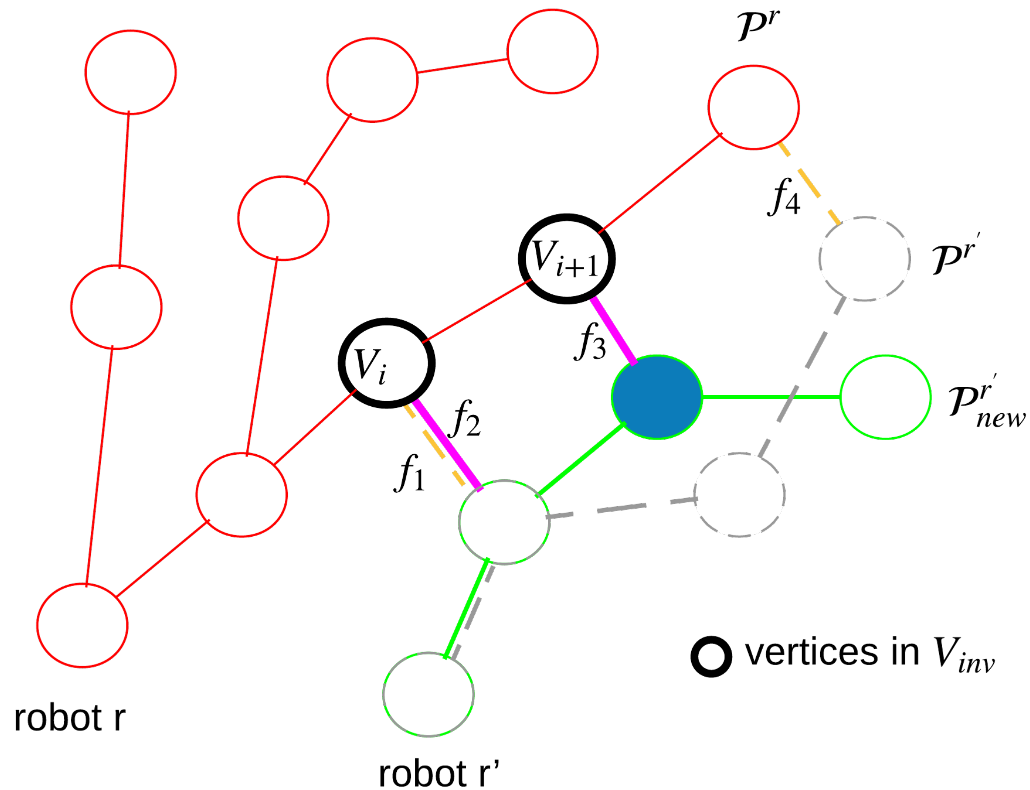
Algorithm Overview – Details

- Iterate **over vertices** in previous and new announced paths
- Identify **involved** vertices in multi-robot factors – collect into set V_{inv}
- Identify and **mark** involved paths from all candidate paths



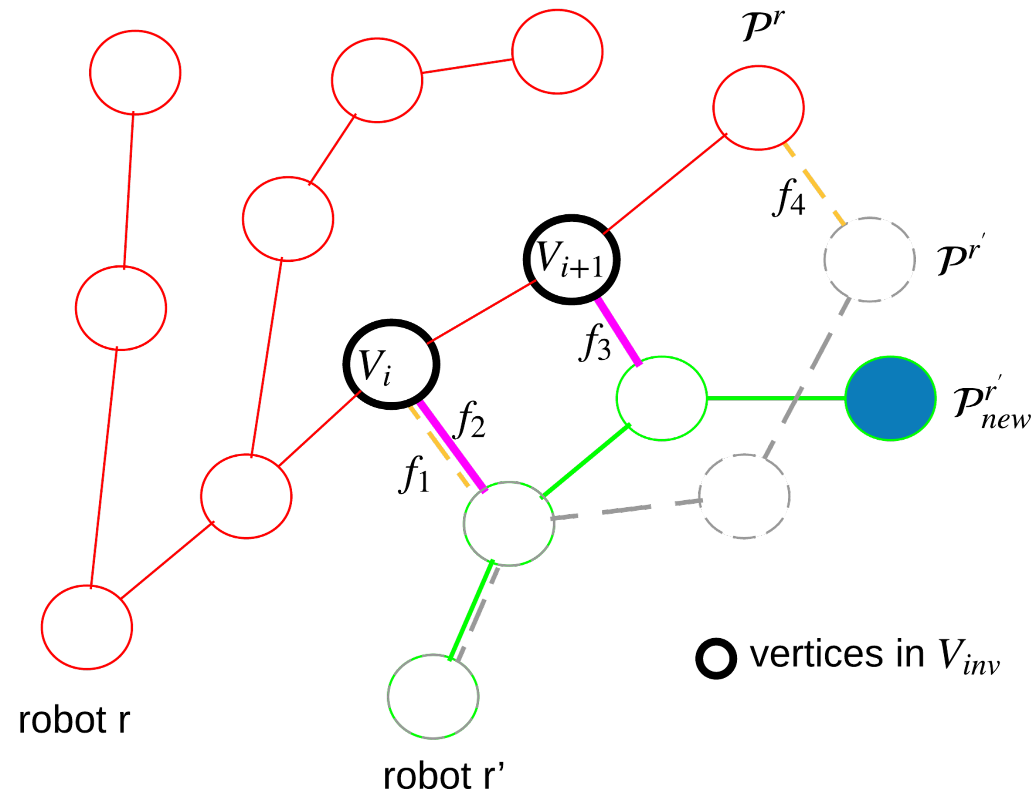
Algorithm Overview – Details

- Iterate **over vertices** in previous and new announced paths
- Identify **involved** vertices in multi-robot factors – collect into set V_{inv}
- Identify and **mark** involved paths from all candidate paths



Algorithm Overview – Details

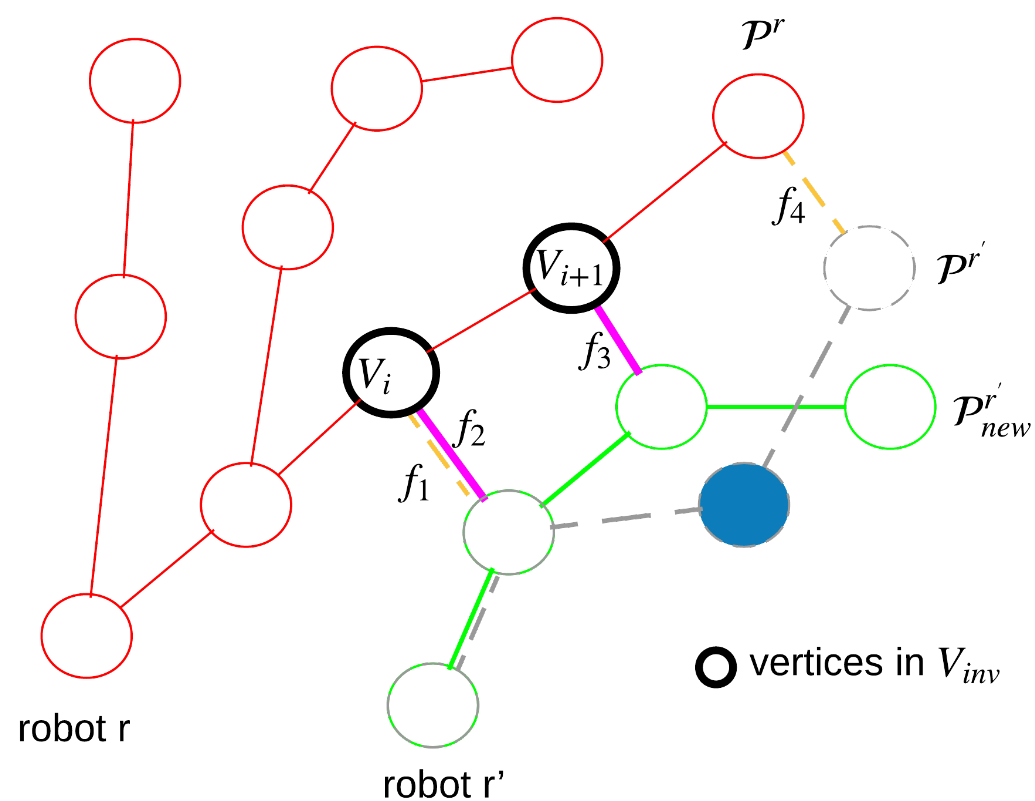
- Iterate **over vertices** in previous and new announced paths
- Identify **involved** vertices in multi-robot factors – collect into set V_{inv}
- Identify and **mark** involved paths from all candidate paths



T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

Algorithm Overview – Details

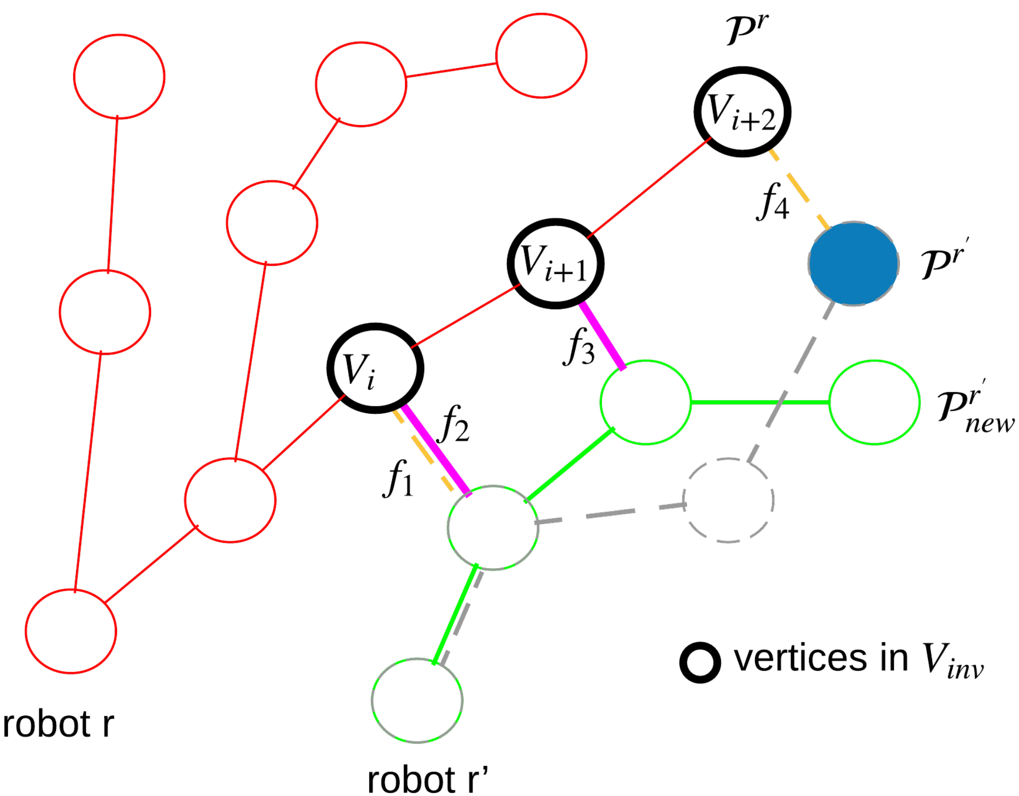
- Iterate **over vertices** in previous and new announced paths
- Identify **involved** vertices in multi-robot factors – collect into set V_{inv}
- Identify and **mark** involved paths from all candidate paths



T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

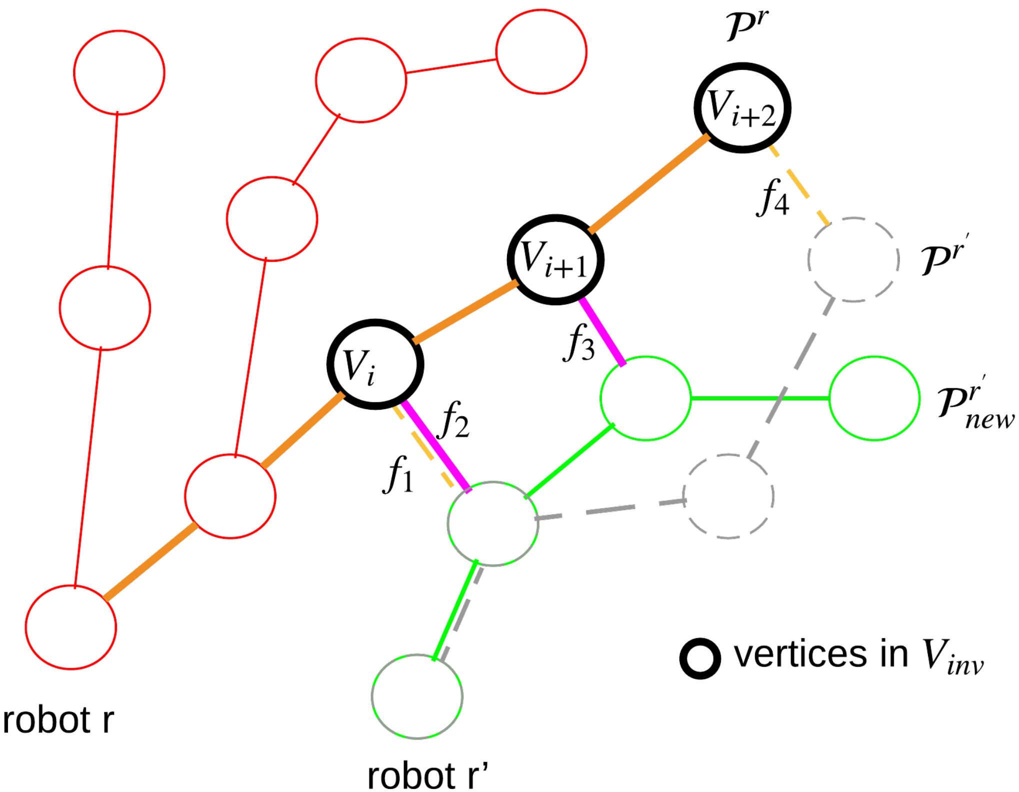
Algorithm Overview – Details

- Iterate **over vertices** in previous and new announced paths
- Identify **involved** vertices in multi-robot factors – collect into set V_{inv}
- Identify and **mark** involved paths from all candidate paths



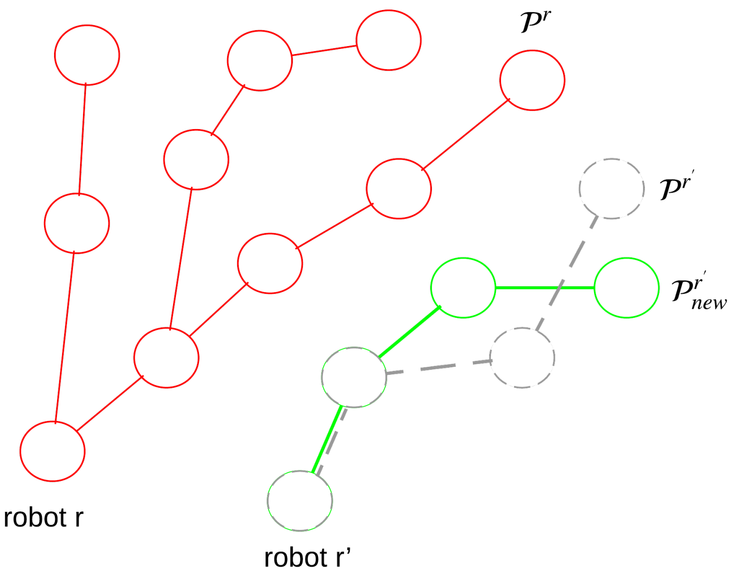
Algorithm Overview – Details

- Iterate **over vertices** in previous and new announced paths
- Identify **involved** vertices in multi-robot factors – collect into set V_{inv}
- Identify and **mark** involved paths from all candidate paths

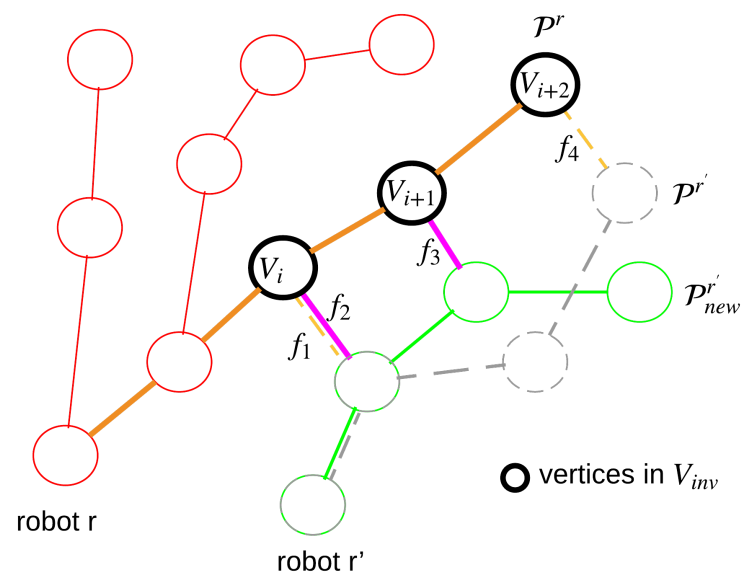


Algorithm Overview

- **Non-marked** paths
  - Calculate once the change $\Delta J^{r'} \doteq \mathbb{E}\left[\sum_{l=1}^{L} \Delta c_l^{r'}\right]$
  - Add $\Delta J^{r'}$ to old objective function



Algorithm Overview



- **Non-marked** paths
  - Calculate once the change $\Delta J^{r'} \doteq \mathbb{E}\left[\sum_{l=1}^{L} \Delta c_l^{r'}\right]$
  - Add $\Delta J^{r'}$ to old objective function
- **Marked** paths
  - Iterate over vertices in V_{inv}, add or remove multi-robot factors
  - Evaluate objective function from new Information matrix

Our method

$$\Lambda'_{k+l} = \Lambda_{k+l} - \sum_{\substack{f\in FG \\ f\notin FG' \\ f.t\le t_{k+l}}} \Lambda(f) + \sum_{\substack{f\in FG' \\ f\notin FG \\ f.t\le t_{k+l}}} \Lambda(f)$$

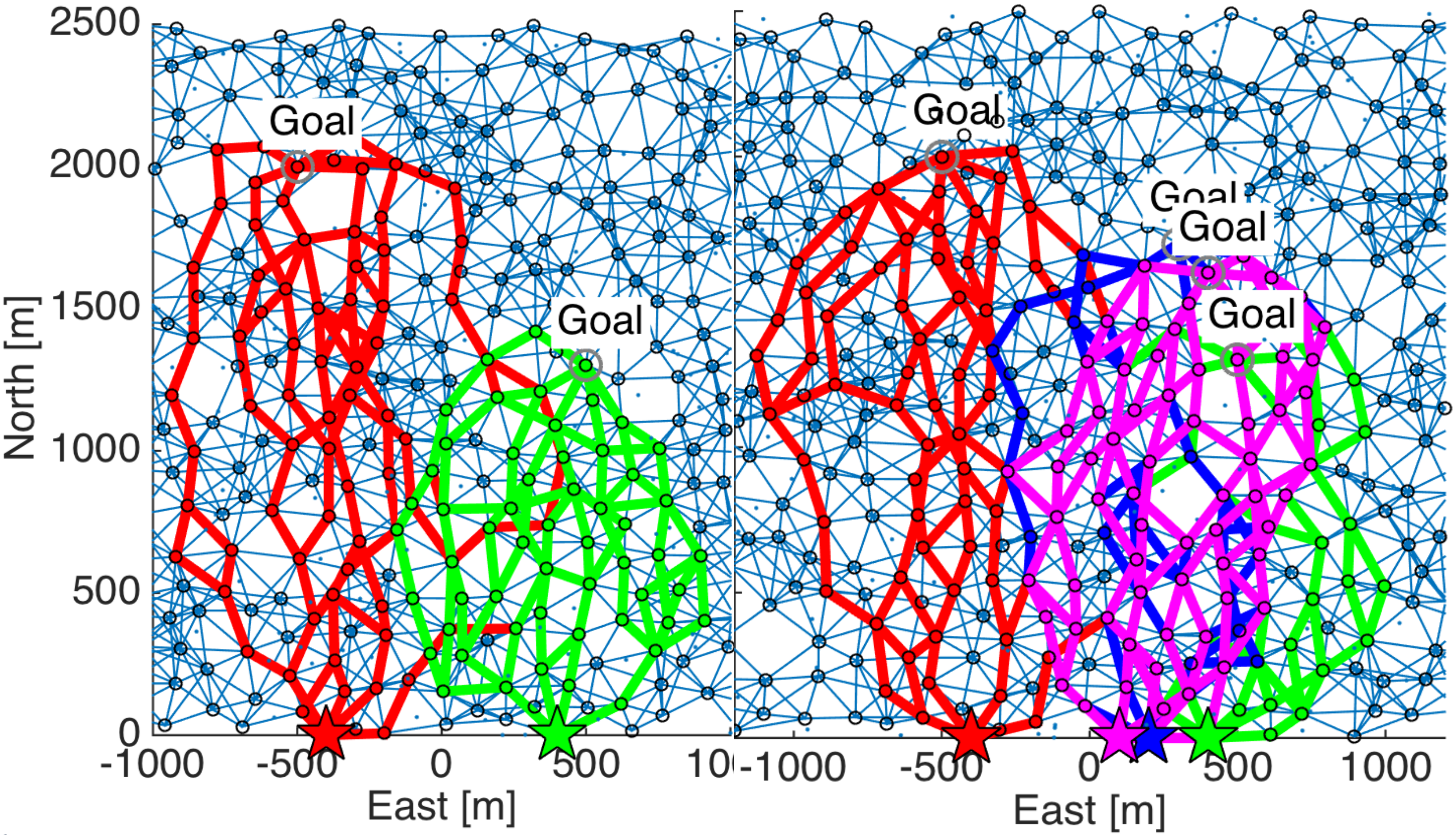
Standard method

$$\Lambda(\mathrm{P}) = \Lambda_k + \sum_{r=1}^{R}\left[\sum_{l=1}^{L(\mathrm{P}^r)} \Lambda_l^{r,local} + \sum_{\{i,j\}} \Lambda_{i,j}^{r,r'}\right]$$

From previous step!

Results

2 robots: 25 candidate paths | **4 robots: 25 candidate paths**

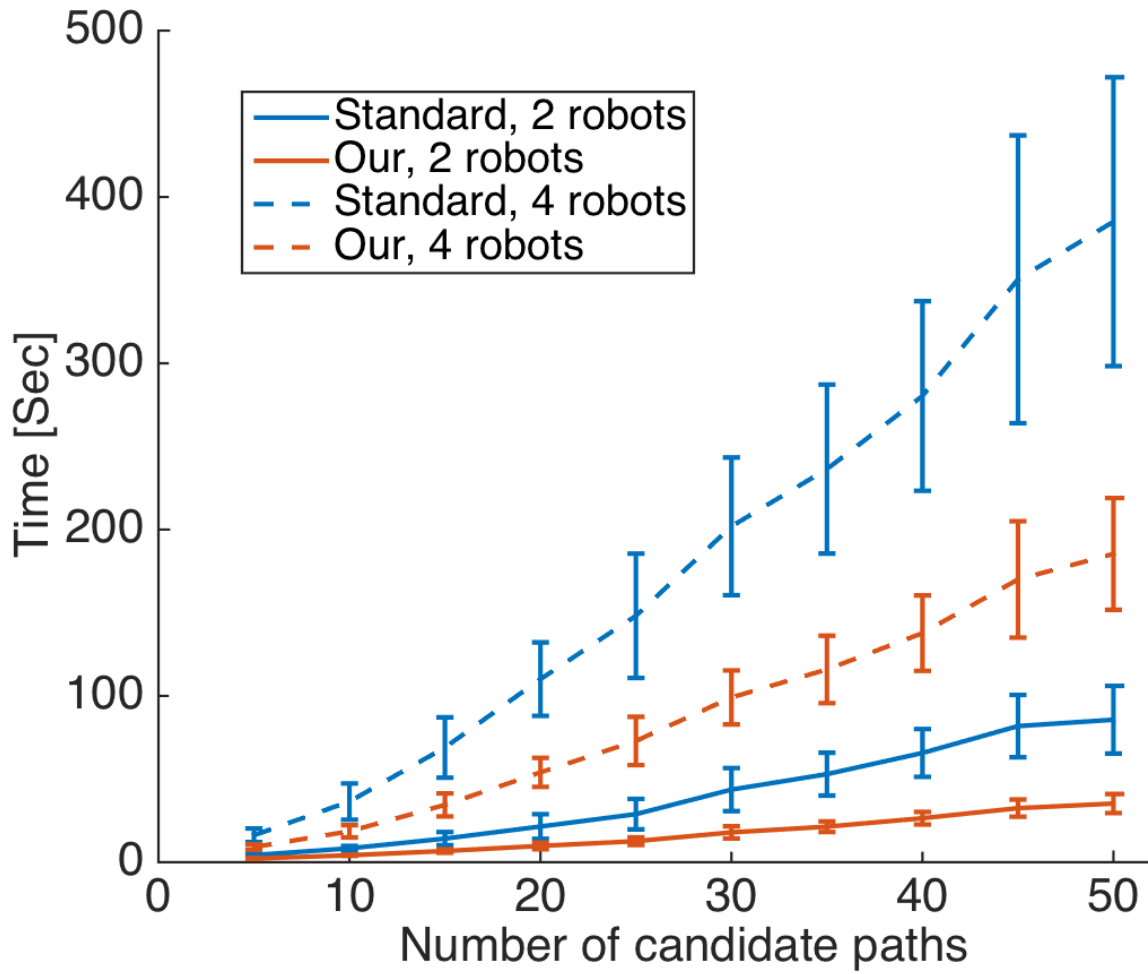


T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

Results

- **Statistical study of 50 runs (2 and 4 robots):**

- **More than twice efficient**
- **Same results as Standard approach**



TECHNION
Israel Institute of Technology

T. Regev and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths, IROS 2016

Conclusions

- Collaborative multi-robot belief space planning in unknown environments
- **Contribution**:
  - Identify impacted paths due to change in announced paths
  - Efficiently re-evaluate belief only for impacted paths
  - One-time re-calculation for all non-impacted paths
  - Performance study in simulation