Distributed Optimization with Pairwise Constraints and its Application to Multi-robot Path Planning

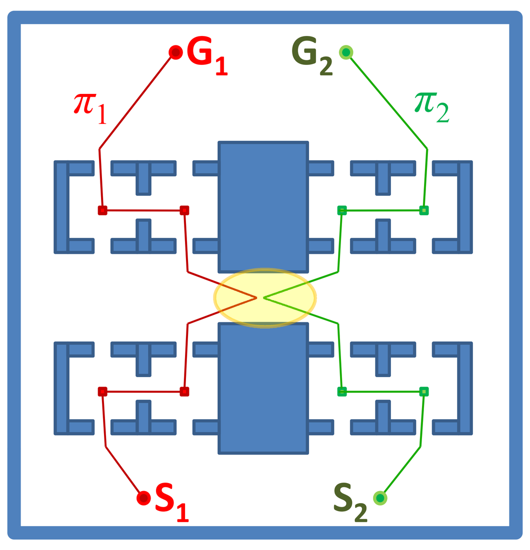
University of Pennsylvania

Subhrajit Bhattacharya Vijay Kumar Maxim Likhachev

GRASP Laboratory

Motivation : Multi-agent Path Planning with Constraints

- Goal directed navigation
  - N heterogeneous robots
    Trajectory of i^{th} robot: π_i
- Intermediate **tasks**
  (e.g., exploration of rooms)
- **Constraints,** $\Omega_{ij}(\pi_i, \pi_j) \le 0$
  (e.g., on time-parametrized distance between trajectories)
- Optimal plan satisfying constraints (minimize net cost)



Problem Definition

Find $\{\pi_1^*, \ldots, \pi_N^*\} = \text{argmin}_{\pi_1 \ldots \pi_N} \sum_{j=1 \ldots N} c_j(\pi_j)$ (1)

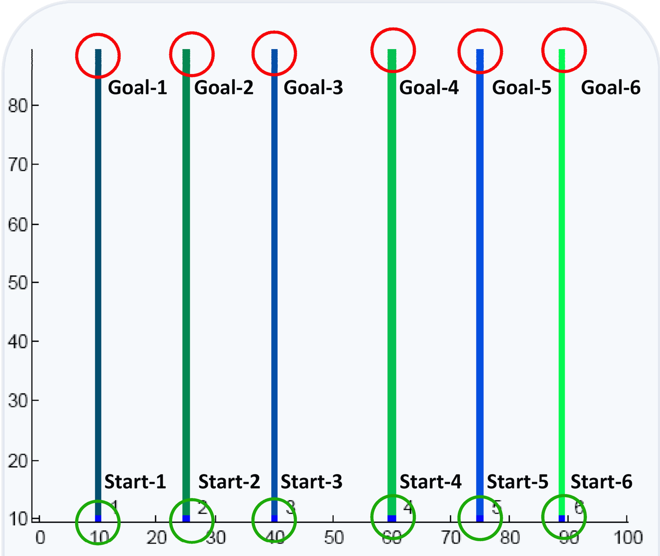
s.t. $\Omega_{ij}(\pi_i^*, \pi_j^*) = 0, \quad i, j = 1 \cdots N$

- Size of joint state space **increases exponentially** with N
  (**coupling** due to constraints)
- Need for **fast planning** as well as **theoretical guarantees**
- Potentially **non-convex** cost and constraint functions
  (e.g., cluttered, non-trivial environments)

Approach : Iterative planning in individual state-space with guarantees

Example: 6 robots with rendezvous constraints

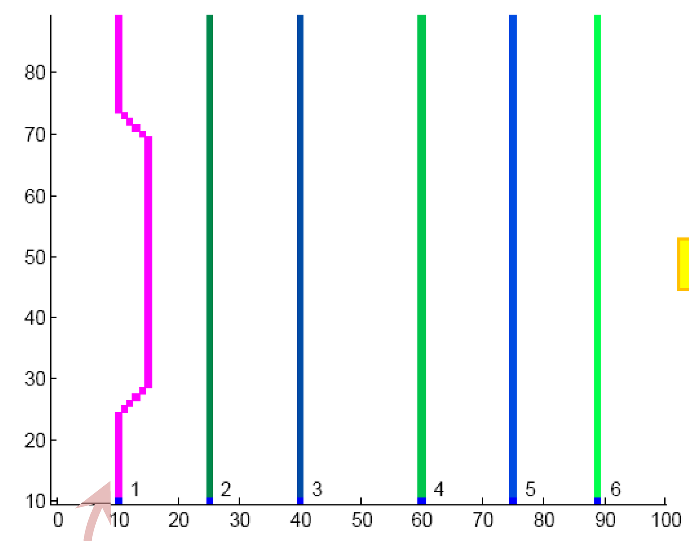
Overview
- A. Convert (1) to a series of unconstrained optimization problems, (2) (soft constraints)
- B. At k^{th} iteration, robot r solves only for π_r based on other robots' current π's
- C. Increase penalty weights on constraint violation



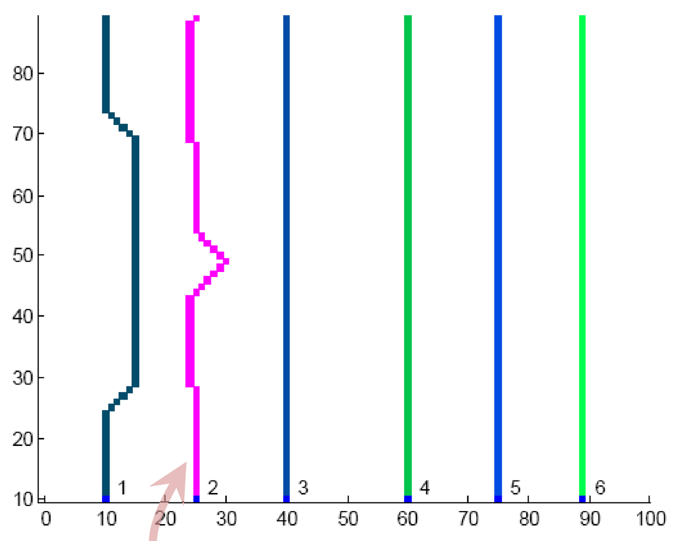
Iteration (*k*) = *0*

$\pi_i^0 = \text{argmin}_{\pi_i} c_i(\pi_i), \quad \forall i \in \mathcal{N}^N$

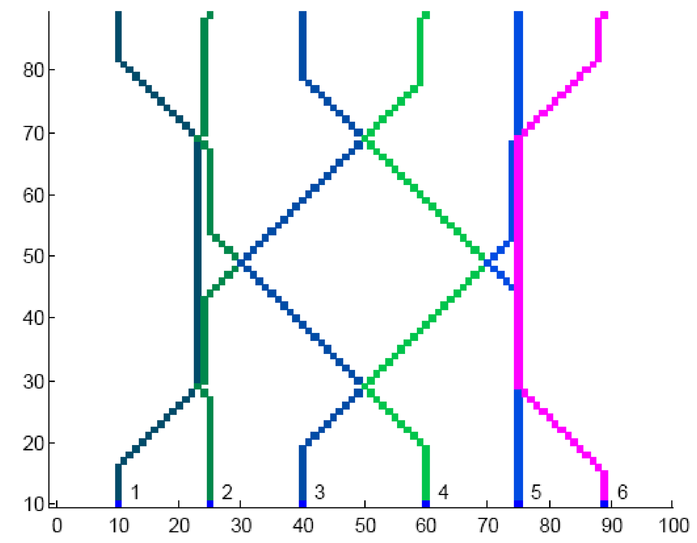
$W_{ij}^0 = 0$



Iteration (*k*) = *1*
Planning robot (*r*) = *1*



Iteration (*k*) = *2*
Planning robot (*r*) = *2*



Iteration (*k*) = *12*

Final converged solution satisfying constraints

$W^{k+1} = W^k + \epsilon^k \, ComputeStepDirection(W^k, \{\pi\}^k, r) \qquad \pi_j^{k+1} = \pi_j^k \ \forall j \neq r$

Subproblem $\pi_r^{k+1} = \text{argmin}_{\pi_r} \left[c_r(\pi_r) + \sum_{i=1 \ldots N,\, i \neq r} W_{ir}^{k+1} \Omega_{ir}(\pi_i^k, \pi_r) \right]$ (2)

Theoretical Analysis

Definitions

$\mathcal{N}^N = \{1, 2, \cdots, N\}$

$\mathcal{P}^N = \{\{1,2\}, \{1,3\}, \cdots, \{1,N\}, \{2,3\}, \{2,4\}, \cdots, \{N-1, N\}\}$

$\mathcal{P}_r^N = \{\{1,r\}, \cdots, \{r-1, r\}, \{r+1, r\}, \cdots, \{N, r\}\}$

V and W are vectors with $N(N-1)/2$ elements

$\{\overline{\Pi}\}(W) := \arg\min_{\{\pi\}} \left[\sum_{k \in \mathcal{N}^N} c_k(\pi_k) + \sum_{\{kl\} \in \mathcal{P}^N} W_{kl} \Omega_{kl}(\pi_k, \pi_l) \right]$

$\Psi_r(W_1, W_2) := \min_{\pi_r} \left[c_r(\pi_r) + \sum_{\{kr\} \in \mathcal{P}_r^N} W_{1,kr} \Omega_{kr}(\overline{\Pi}_k(W_2), \pi_r) \right]$

For a small ϵ , V is a ***Separable Optimal Flow Direction*** for Ψ_r at W iff: $\Psi_r(W + \epsilon V, W) - \Psi_r(W, W) \le \Psi_r(W + \epsilon V, W + \epsilon V) - \Psi_r(W, W + \epsilon V)$
$\Rightarrow (\epsilon V)^T \left[\Psi_r^{(1,1)}(W, W) \right] (\epsilon V) \ge 0$
and, $V_{ij} = 0, \quad \forall \{i,j\}$ such that $r \notin \{i,j\}$

V is an ***Ascent Direction*** at W iff: $\sum_{\{ij\} \in \mathcal{P}^N} V_{ij} \Omega_{ij}(\overline{\Pi}_i(W), \overline{\Pi}_j(W)) > 0$

Theorem 1: If the *Step Direction* returned by procedure *ComputeStepDirection* at the k^{th} iteration of the Algorithm, along with a small step size, ϵ^k, define a *Separable Optimal Flow* at W^k for Ψ_{r_κ}, $\forall\, k$, then $\forall\, k$
$\{\pi_1^k, \ldots, \pi_N^k\} = \arg\min_{\{\pi\}} \left[\sum_{i \in \mathcal{N}^N} c(\pi_i) + \sum_{\{ij\} \in \mathcal{P}^N} W_{ij}^k \cdot \Omega_{ij}(\pi_i, \pi_j) \right]$

There **exists directions**, which we call *Separable Optimal Flow Directions,* in which **we can increment the penalty weight vector *W***, such that the global optimum for the new set of penalty weights differs from the previous global optimum in only one partition of the optimization variables, namely π_r. Thus, by moving along such a direction in k^{th} iteration, **we only need to change π_{r_k}, and still remain at an optimum of the penalized net cost**. *i.e.* $\pi_i^k = \overline{\Pi}_i(W^k), \quad \forall i, k$

Theorem 2: If the condition in Theorem 1 holds, and the *Step Direction* returned by procedure *ComputeStepDirection* at the k^{th} iteration of the Algorithm is also an *Ascent Direction* at W^k, for all k, then the Algorithm converges to an optimal solution, if one exists.

If we always increment the penalty weights along directions that are both *Ascent Directions* and *Separable Optimal Flow Directions*, we will eventually converge to the global optimum, if it exists.

Theorem 3: If the functions c_r and Ω_{ij} are differentiable up to second order, and $\Omega_{ij}(\pi_i, \pi_j)$ is of the form $G_{ij}(\pi_i - \pi_j)$, where G_{ij} is continuous, smooth and even, then we can compute a *Step Direction*, if one exists, that satisfy Theorems 1 & 2, at a given W^k. We use mollification techniques to smoothen c_r and Ω_{ij} if and when required.

Results

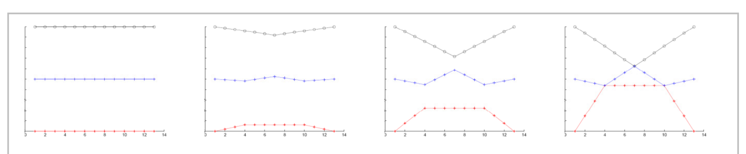
An exact implementation:

$\pi_r = [start_r, y_{r2}, y_{r3}, \cdots, y_{rL}, goal_r]^T$

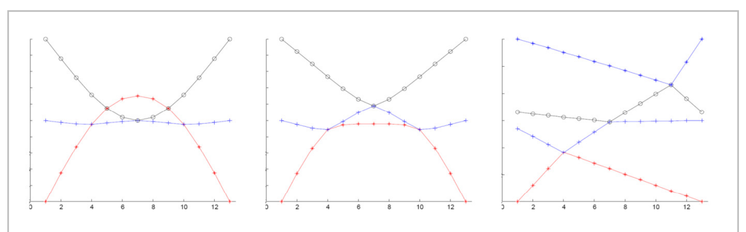
$c_r(\pi_r) = \alpha \left((y_{r2} - start_r)^2 + (y_{r3} - y_{r2})^2 + \cdots \right)^{1/2} + \beta \left(((y_{r4} - y_{r3}) - (y_{r3} - y_{r2}))^2 + ((y_{r5} - y_{r4}) - (y_{r4} - y_{r3}))^2 + \cdots \right)$

$\Omega_{ab}(\pi_a, \pi_b) = \left((y_{ac_1} - y_{bc_1})^2 + (y_{ac_2} - y_{bc_2})^2 + \cdots \right)^{1/2}$

Planning in continuous space with rendezvous constraints



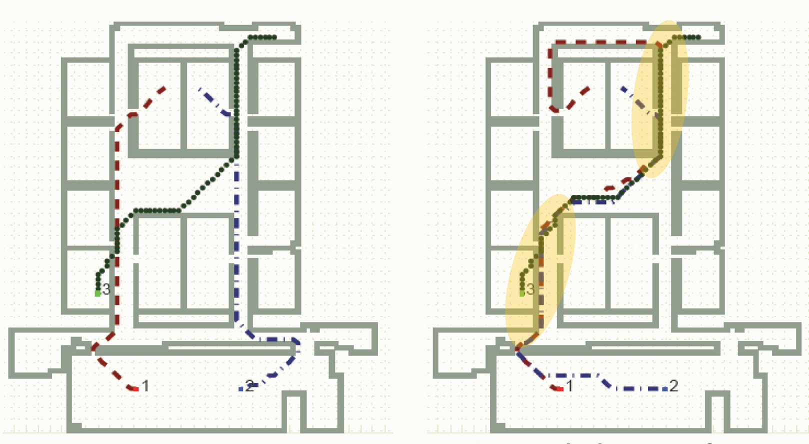
(a) Unconstrained plans, $k = 0$ (b) $k = 50$ (c) $k = 150$ (d) Converged solution, $k = 216$



(a) Symmetric case: $\alpha = 0, \beta = 1$ (b) Symmetric case: $\alpha = 1, \beta = 0.2$ (c) An asymmetric case: $\alpha = 1, \beta = 0$

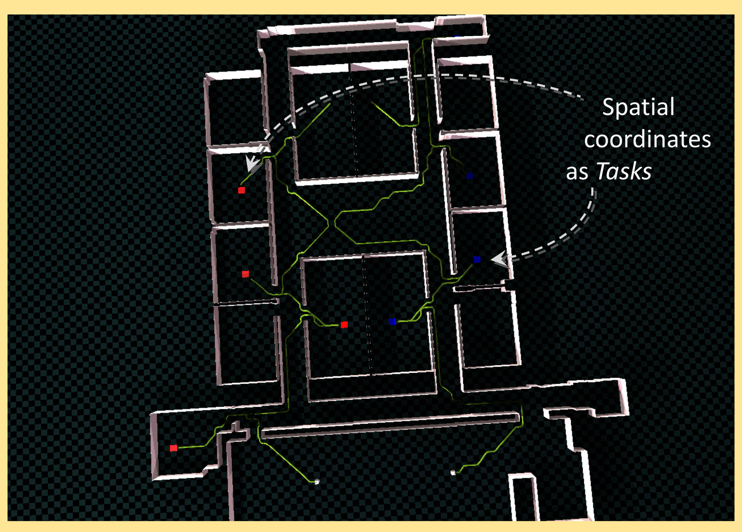
Planning in a cluttered environment with spatio-temporal constraints:

***Ex. 1:* Planning with extended rendezvous**

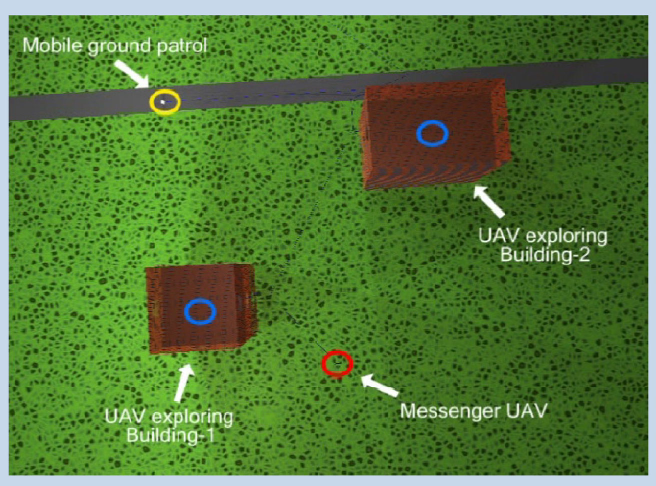


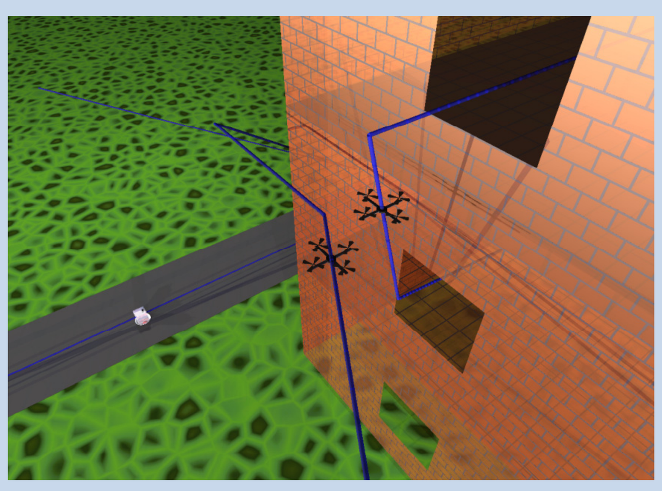
Unconstrained solution Optimal plan satisfying communication constraints

***Ex.2:* Planning with tasks and constraints**



***Ex.3:* Heterogeneous agents performing complex tasks in 3D:**





Conclusions

- Developed an algorithm for efficiently solving large optimization problems with nonlinear constraints in distributed fashion.
- Theoretical analysis gives conditions required for guarantees on convergence and optimality.
- Implemented the algorithm on multi-robot planning problems in complex environments.

Acknowledgements

We gratefully acknowledge support from ONR grant no. N00014-09-1-1052, NSF grant no. IIS-0427313, ARO grant no. W911NF-05-1-0219, ONR grants no. N00014-07-1-0829 and N00014-08-1-0696, and ARL grant no. W911NF-08-2-0004.