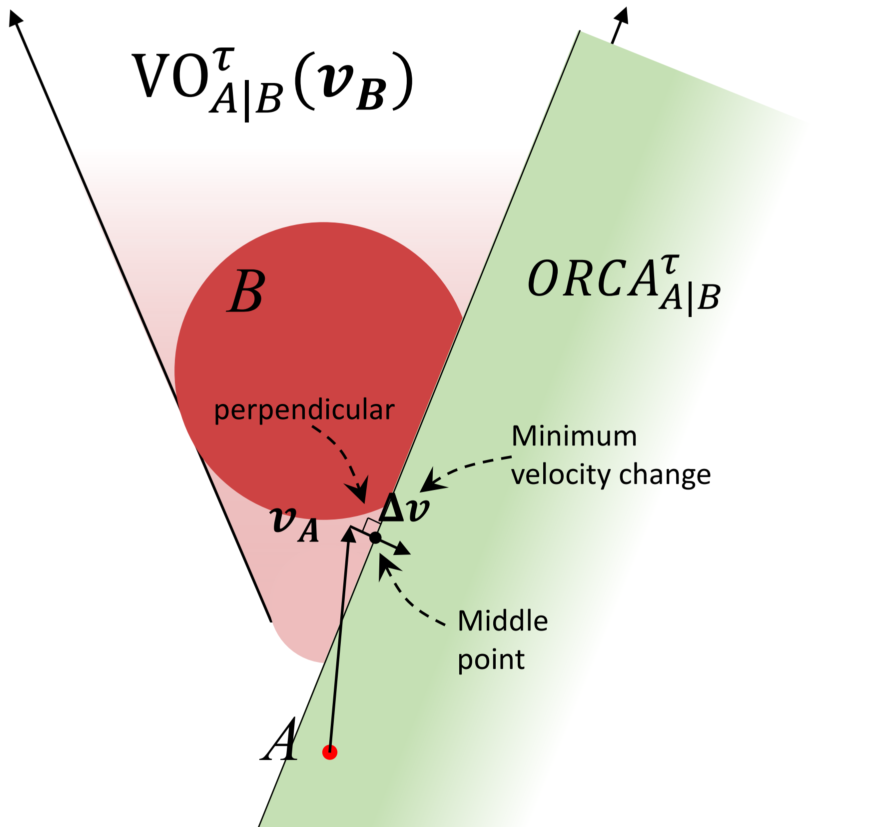
Optimal Reciprocal Collision Avoidance (ORCA)

The Main Idea: Each agent takes just **half** the responsibility.

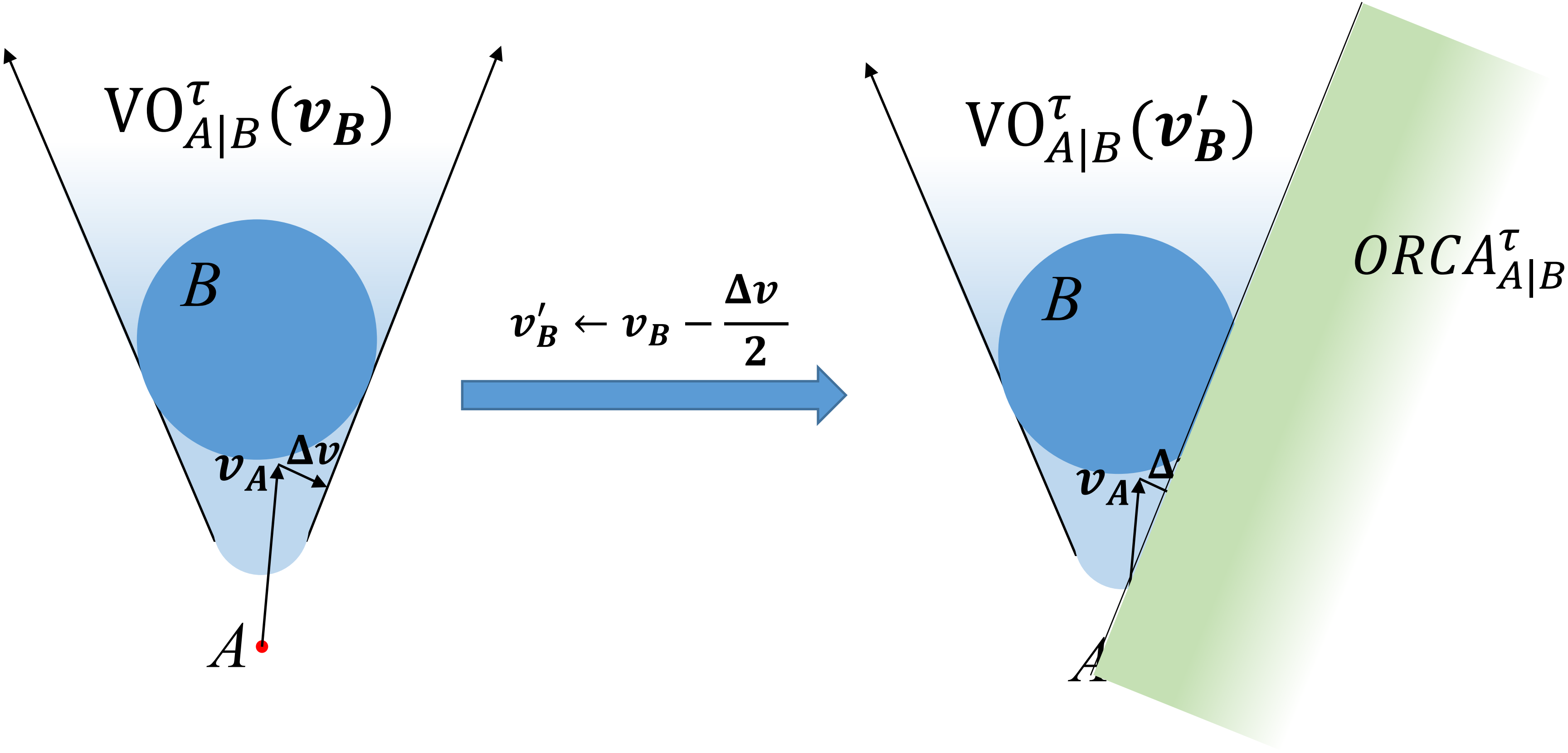
Construction of $ORCA_{A|B}^{\tau}$: see the demonstration.

Selection of optimal velocity: the optimal velocity change for A is $\frac{1}{2}\Delta v$, for B is $-\frac{1}{2}\Delta v$.

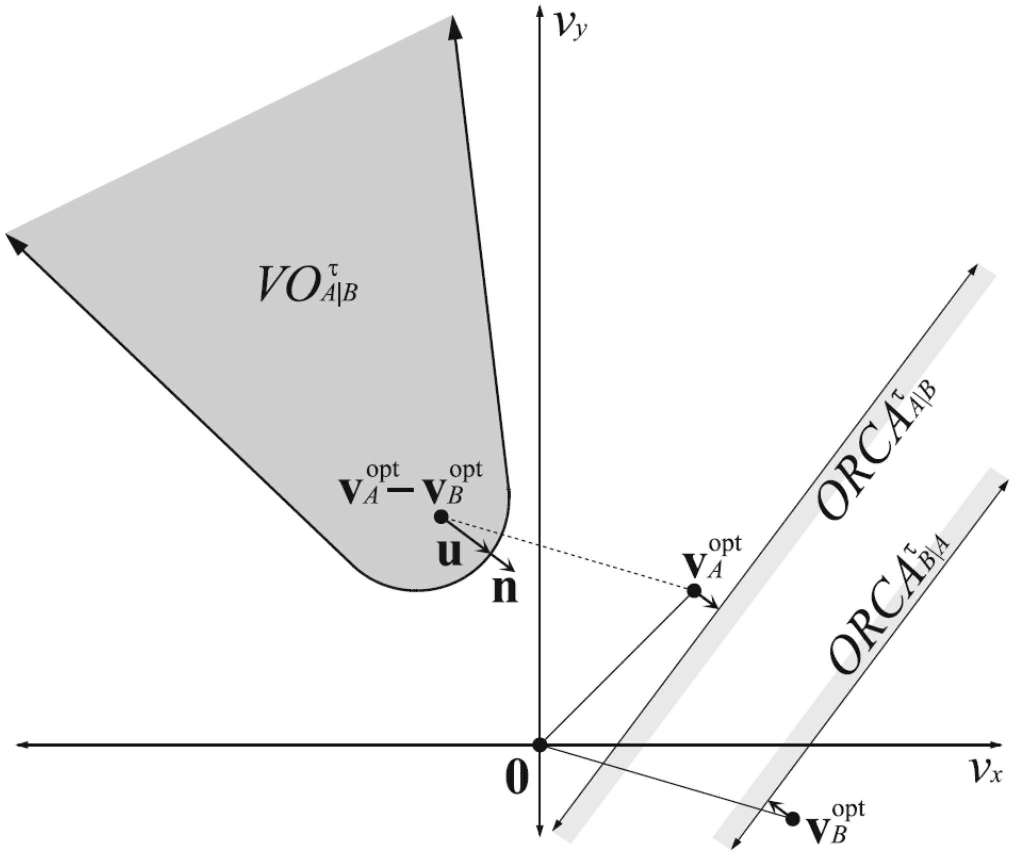


Optimality definition and proof can be found in pages 6-9 of the paper:
Berg, Jur van den, et al. "Reciprocal n-body collision avoidance." Robotics research. Springer, Berlin, Heidelberg, 2011. 3-19.

Optimal Reciprocal Collision Avoidance (ORCA)



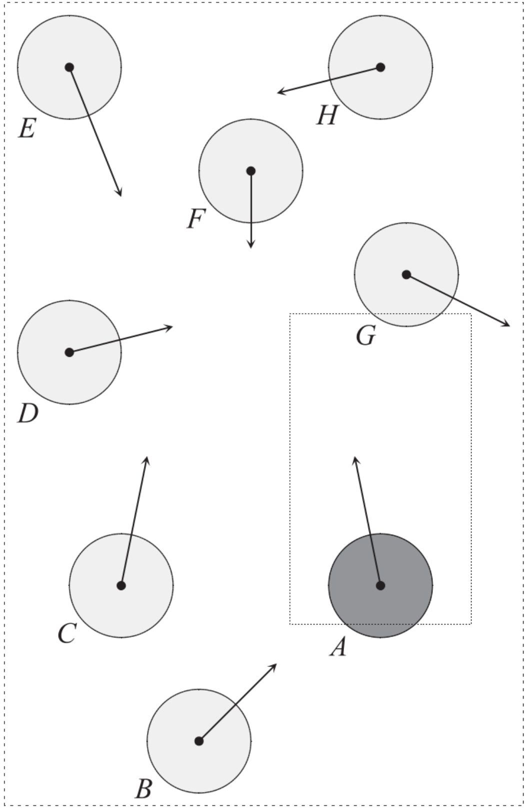
Optimal Reciprocal Collision Avoidance (ORCA)

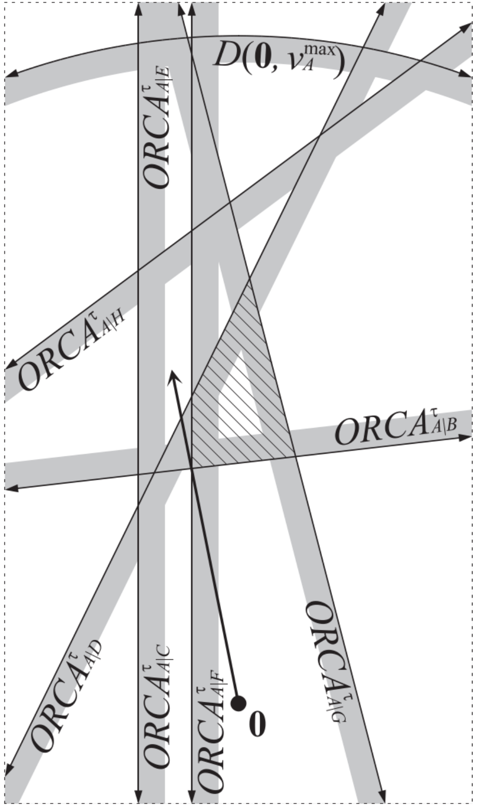


Berg, Jur van den, et al. "Reciprocal n-body collision avoidance." Robotics research. Springer, Berlin, Heidelberg, 2011. 3-19.

Optimal Reciprocal Collision Avoidance (ORCA)

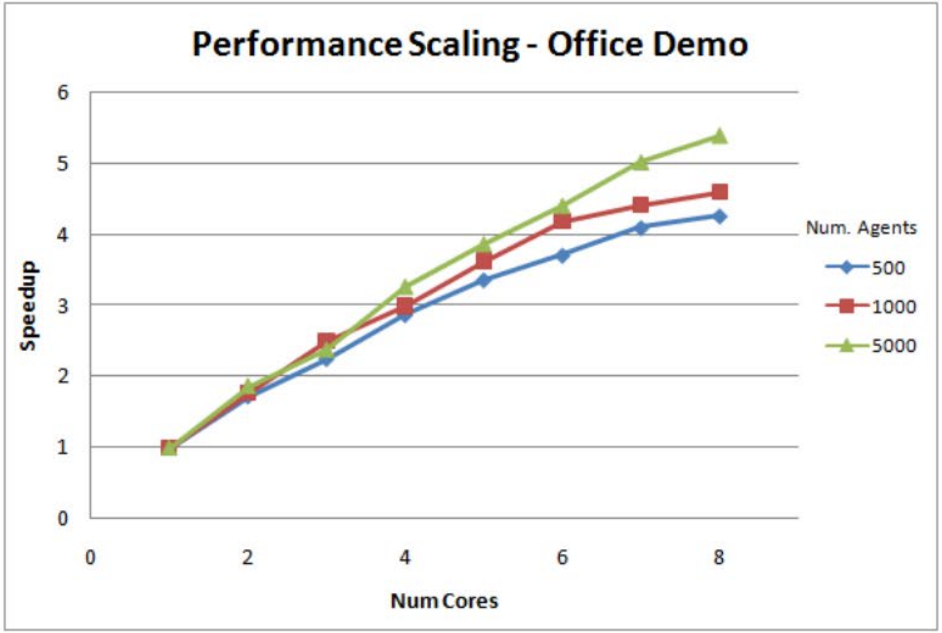
Multiple Obstacle Case

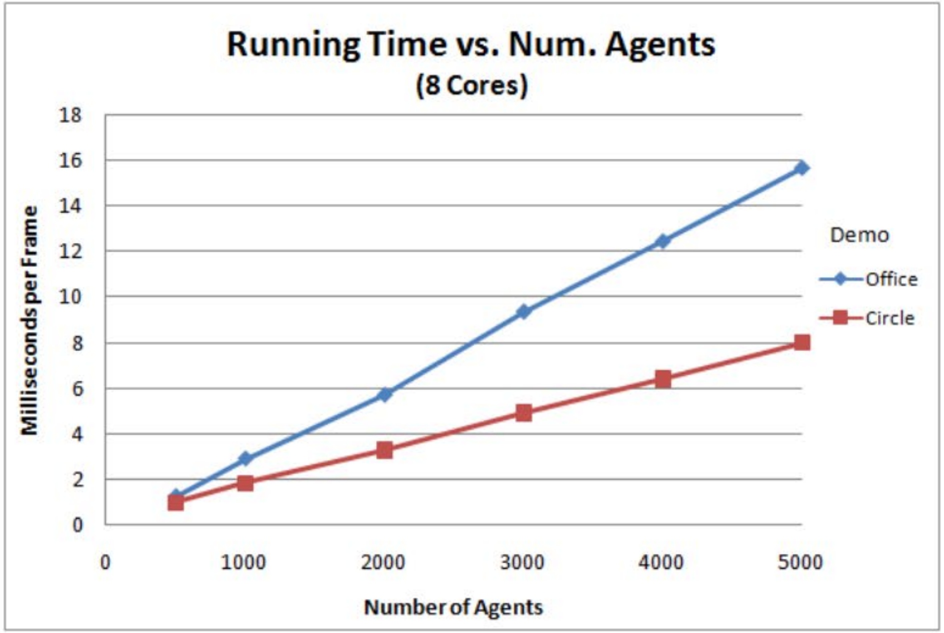




Berg, Jur van den, et al. "Reciprocal n-body collision avoidance." Robotics research. Springer, Berlin, Heidelberg, 2011. 3-19.

Optimal Reciprocal Collision Avoidance (ORCA)

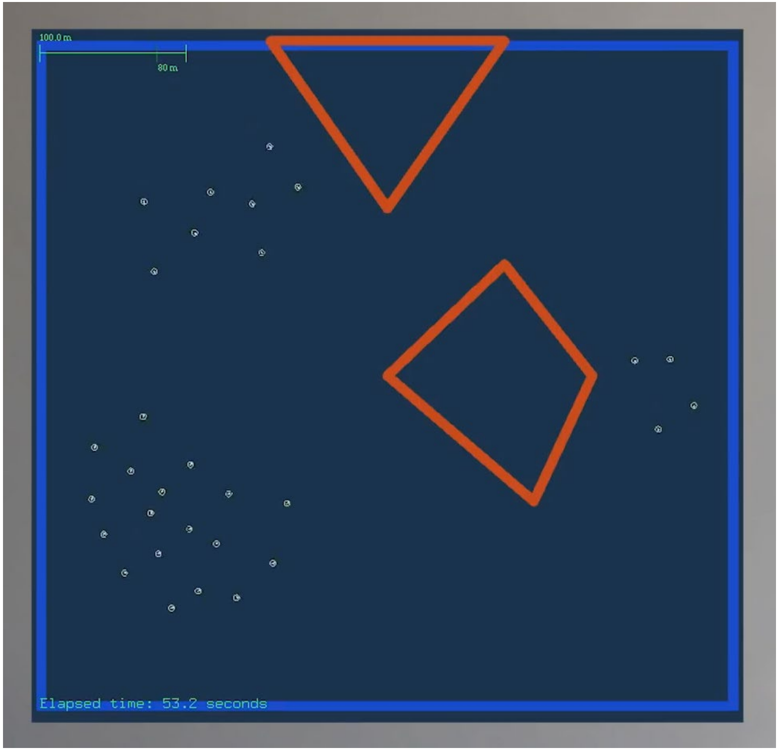




Berg, Jur van den, et al. "Reciprocal n-body collision avoidance." Robotics research. Springer, Berlin, Heidelberg, 2011. 3-19.

Contents

Background



[1] Vásárhelyi, Gábor, et al. "Optimized flocking of autonomous drones in confined environments." *Science Robotics* 3.20 (2018): eaat3536.

The Main Idea

The main idea: In order to fly like a flock of birds, each agent is controlled by three forces:

- Short-term: repulsive force $\mathbf{v}^{rep}$ to neighbors and obstacles;
- Medium-term: velocity alignment force $\mathbf{v}^{frict}$ that aligns the movement with neighbors;
- Long-term: **attractive force to the goal** $\mathbf{v}^{flock}$;

The total force is the combination of above forces:

$$\mathbf{v}^{exe} = \mathbf{v}^{rep} + \mathbf{v}^{frict} + \mathbf{v}^{flock}.$$

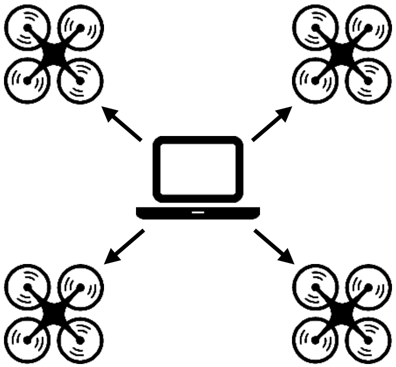
Advantages: this strategy mimics the movements of natural animal community well.
Disadvantages: the performance is sensitive to parameter settings.
Candidate solutions: automatic parameter tuning using evolutionally strategies[1].

[1] Vásárhelyi, Gábor, et al. "Optimized flocking of autonomous drones in confined environments." *Science Robotics* 3.20 (2018): eaat3536.

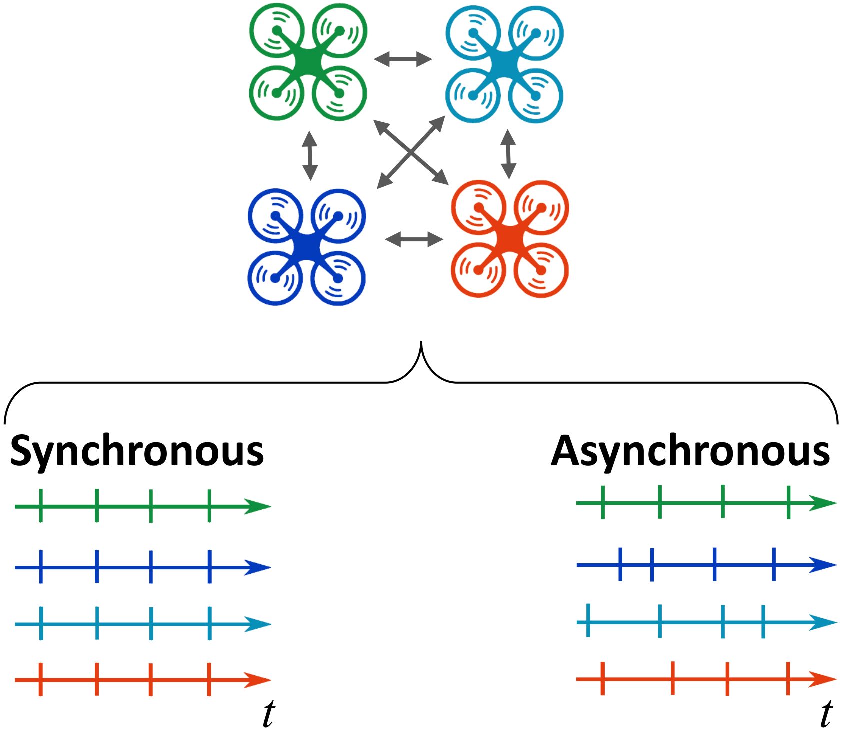
Contents

System Architecture

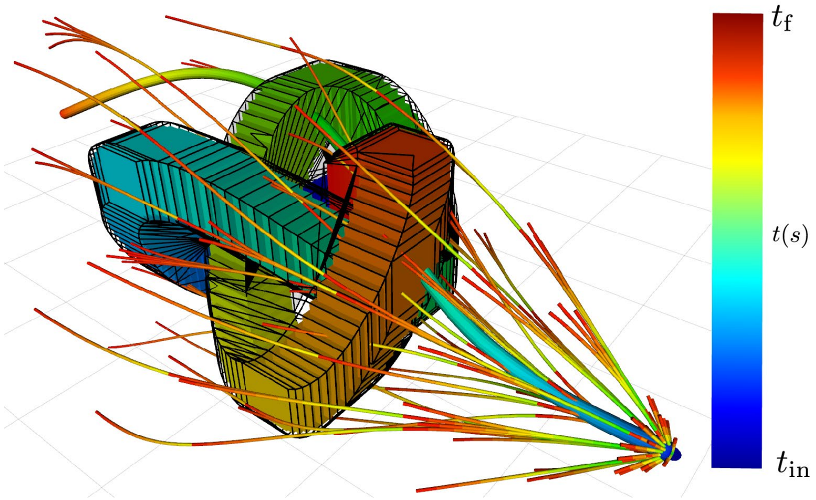
Centralized

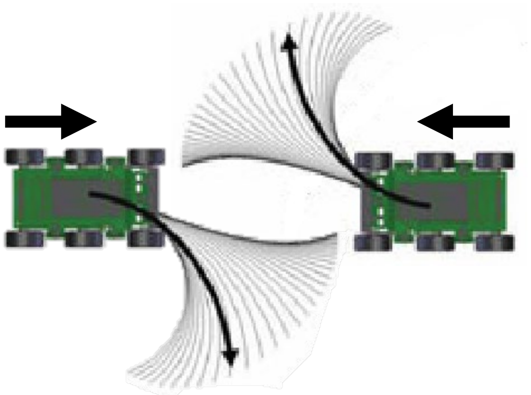


Decentralized



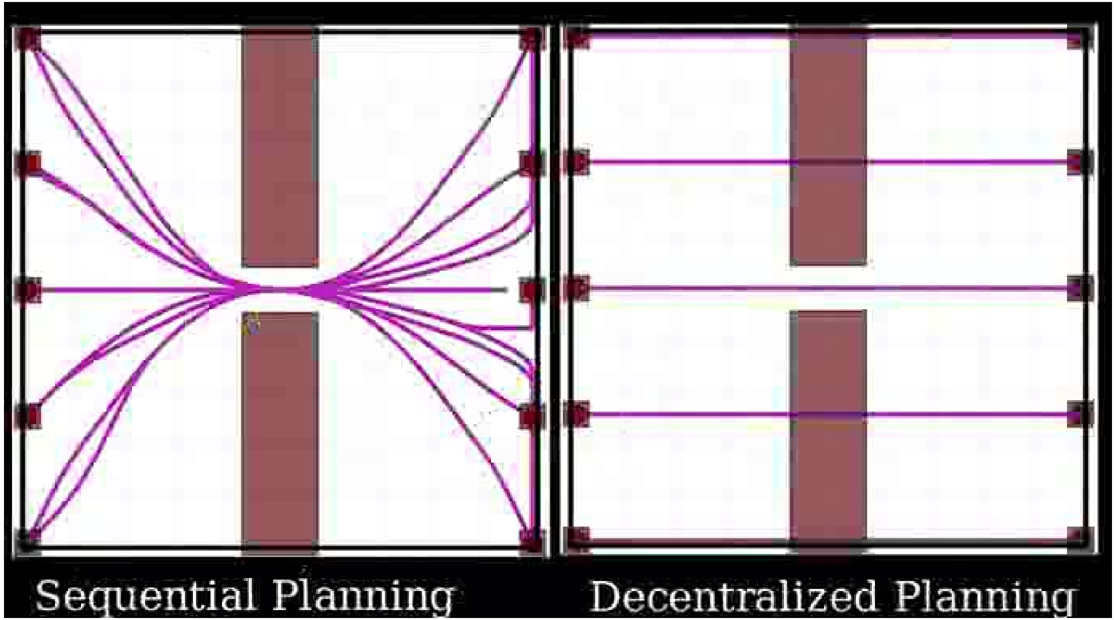
Search/Sampling Based





Hybrid A*, RRT*, etc.

Search/Sampling Based



Plan only once for each agent

Repeated planning

∴ generally used in ordered planning

Liu, Sikang, Kartik Mohta, Nikolay Atanasov, and Vijay Kumar. "Towards search-based motion planning for micro aerial vehicles." arXiv preprint arXiv:1810.03071 (2018).

Optimization Based

The Basic Formulation

Optimization-based planning rooted in optimal control theory formulates trajectory planning as

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) \\ \text{s.t. } &\boxed{\mathcal{G}(\mathbf{x}) \leq \mathbf{0},} \\ &\mathcal{H}(\mathbf{x}) = \mathbf{0} \end{aligned}$$

where $\mathbf{x}$ are trajectory parameters or actuator commands.

Standard reciprocal collision avoidance penalty: avoiding getting too close to each other at any timestamp. One of the basic formulations:

$$d_{i,j}(t) = |p_i(t) - p_j(t)|,$$
$$\forall\, t \in [T_0, T_m], \mathcal{C} - d_{i,j}(t) \leq 0.$$

$p_i(t), p_j(t)$: the positions of agent i and j at time t, respectively;
$\mathcal{C}$: the minimum allowed clearance between two agents.

Optimization Based

The Basic Formulation

How to formulate this part?

Standard reciprocal collision avoidance penalty: avoiding getting too close to each other at any timestamp. One of the basic formulations:

$$d_{i,j}(t) = |p_i(t) - p_j(t)|,$$
$$\forall\, t \in [T_0, T_m], \mathcal{C} - d_{i,j}(t) \leq 0.$$

$p_i(t), p_j(t)$: the positions of agent i and j at time t, respectively;

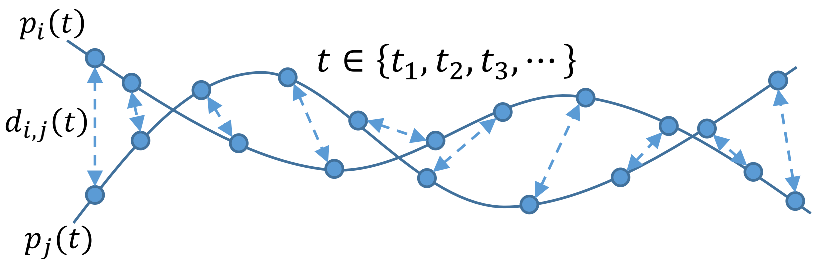
$\mathcal{C}$: the minimum allowed clearance between two agents.

Optimization Based

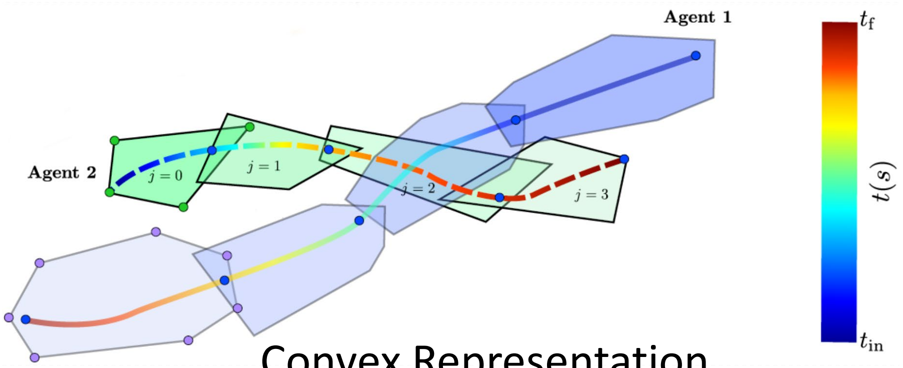
Reciprocal Avoidance Formulation

$$d_{i,j}(t) = |p_i(t) - p_j(t)|,$$
$$\forall\, t \in [T_0, T_m], \mathcal{C} - d_{i,j}(t) \le 0.$$

- It contains infinite number of constraints as it holds for every timestamp



Discretization
- Less Safe
- Computationally Efficient (Simple)
- Less Conservative



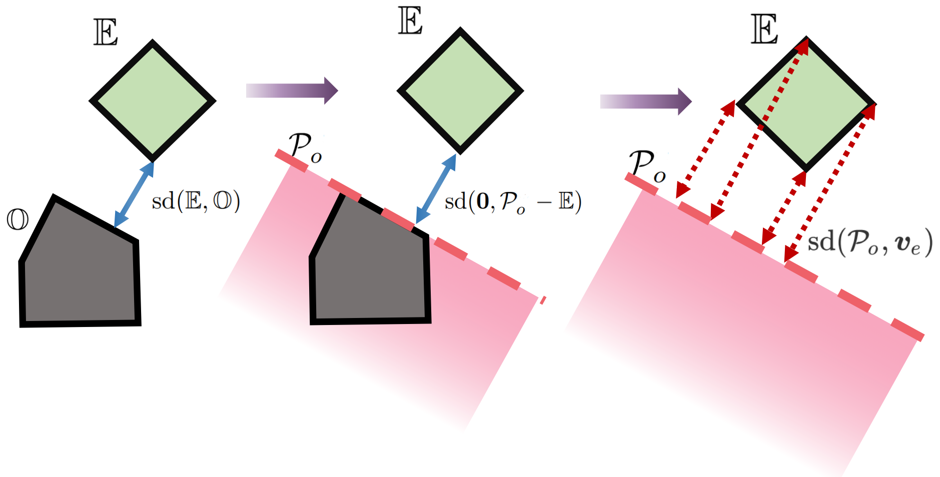
Convex Representation
- Safe
- Computationally Expensive (Complex)
- Conservative

Optimization Based

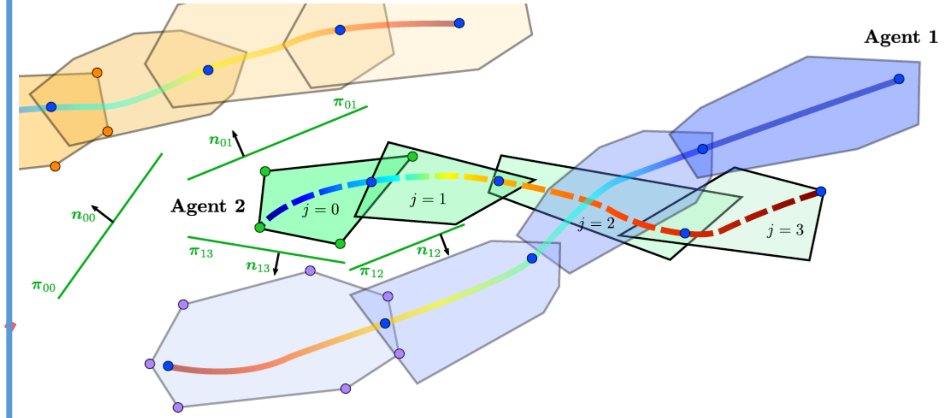
More about Convex Representation

Avoidance constraints between convex polyhedrons are nonconvex.

∴ We convert these constraints into distances between vertexes and planes.



Find a plane $\mathcal{P}_o$ on $\mathbb{O}$ with the closest distance to vertexes on $\mathbb{E}$



Optimize planes separating every two planes