Multi-agent Trajectory Optimization with Collision Avoidance using Smooth, Non-convex Constraints

Anonymous author(s)

Association for the Advancement of Artificial Intelligence
2275 East Bayshore Road, Suite 160
Palo Alto, California 94303
publications22@aaai.org

Abstract

In the domain of trajectory optimization for multiple agents, collision avoidance poses a challenge to conventional solutions, since the introduction of collision avoidance constraints can cause the solution curve to become non-differentiable, which causes issues for gradient-based optimizers. One common approach to deal with this is to relax the problem into an approximate formulation, by making significant simplifications of the problem formulation. In this work, we propose an optimization-based method for collision avoidance for multi-agent motion planning, without convexification or linearization of the agent dynamics. The method is validated in simulation using two full-dimensional mobile robots, modeled with non-linear dynamics, where an optimal trajectory is found while avoiding a collision. Additionally, it is evaluated on a similar scenario, with some simplifications that allow it to also be solved using mixed-integer programming. The results show that the new method generates an equivalent motion plan and that the algorithm has a similar time complexity as the mixed-integer program, while not necessarily being subject to the same restrictions on the modeling.

1 Introduction

Motion planning techniques for autonomous vehicles and mobile robotic systems is a research area in which interest has been growing for the past years (Goerzen, Kong, and Mettler 2010; Paden et al. 2016; Mohanan and Salgoankar 2018). As the number of autonomous systems increases, and low-latency, high-performance computing systems continue to improve, multi-agent motion planning with collision avoidance has become one of the emerging topics (Stern et al. 2019; Andreychuk et al. 2019).

In this work, we consider optimization-based methods for multi-agent motion planning with collision avoidance. In such a framework, the area-avoidance problem becomes an inherently non-convex constraint in the problem formulation. Many solutions approximate obstacles or other agents by convex methods that can be solved with high efficiency and accuracy (Ong and Gerdes 2015; Li et al. 2020). However, this can also make the solution overly conservative, or impose limitations on the planning horizon, which can lead to local infeasibility.

Another more widely deployed method is the mixed-integer programming (MIP) approach (Prodan et al. 2010; Burger and Lauer 2018; Liu et al. 2020). By the introduction of binary choices into the decision vector, the convex problem becomes a combinatorial one, which is solved using branch and bound. Despite the resulting complexity being high, and the limitations of linear modeling, there are high-performance commercial MIP solvers available, which gives the method its popularity.

By modeling obstacles and other vehicles as ellipses, (Alrifaee, Kostyszyn, and Abel 2016) demonstrates that Lagrange relaxation can be used to solve the collision avoidance problem in a lane-keeping context, where velocities are to be held constant.

The contribution of this work is the proposal of a non-convex, optimization-based method for collision avoidance for multi-agent motion planning, without convexification or simplification of the agent dynamics. Building on (Soloperto et al. 2019; Zhang, Liniger, and Borrelli 2021), we extend the notion to multi-agent planning. We show that by introducing the dual of a set-projection subproblem, it is possible to re-write collision avoidance specifications as a set of smooth, differentiable constraints, suitable for general non-linear numerical solvers. The method is evaluated in simulation for a two-agent scenario with full-dimensional, non-linear models, and evaluated against the MIP approach in a simplified, linear test scenario.

In this work, we assume that all agents can be modelled as convex polygons, which is a close approximation for many kinds of mobile robots and autonomous vehicles.

Notation

A vector x is assumed to be a column unless explicitly stated otherwise. By $x^\top$ is meant the transpose of the vector. We denote by $x(i)$ the i^{th} component of the vector. When $x(i) \geq y(i)$, for all components i of two vectors x, y of equal length, it is denoted as $x \succeq y$. Without any loss of clarity, the same notation is used for the case where y is scalar. For example, $x \succeq 0$ implies that each component of x is non-negative. We use $\mathbb{R}$ to denote the set of real numbers and $\mathbb{N}$ for the set of non-negative integers. The ℓ_p matrix norm is written as $\|x\|_p$, where $p = 1, 2, \ldots, \infty$. The $m \times m$ identity matrix is written as I_m, and $\mathbf{1}$ denotes an appropriately sized vector containing only ones.

2 Problem Formulation

In this work, we consider the centralized control problem of $M \in \mathbb{N}$ heterogeneous agents, where the goal is to satisfy individual motion specifications while minimizing some objective function, without any collisions. We define the set of agents $\mathcal{M} = \{1, \ldots, M\}$. In particular, we are concerned with discrete-time trajectory optimization programs over time-instances $k = 1, \ldots N$, expressed as

$$\underset{\boldsymbol{u}}{\text{minimize}} \sum_{k=1}^{N} J(x_k, u_k), \tag{1}$$

$$\text{subject to } x_0 = x_{\text{init}},\ x_N = x_{\text{des}} \tag{2}$$

$$x_{k+1} = f(x_k, u_k), \tag{3}$$

$$g(x_k, u_k) \preceq 0, \tag{4}$$

$$\mathcal{X}(x_k^{(i)}) \cap \mathcal{X}(x_k^{(j)}) = \emptyset, \quad j > i. \tag{5}$$

A detailed description of the problem is given below.

For each agent $j \in \mathcal{M}$, its state and input vectors are written as $x^{(j)} \in \mathbb{R}^{n_j}$ and $u^{(j)} \in \mathbb{R}^{m_j}$, respectively. Their individual dynamics are given by $x_{k+1}^{(j)} = f^{(j)}(x_k^{(j)}, u_k^{(j)})$, where $f^{(j)} : \mathbb{R}^{n_j} \times \mathbb{R}^{m_j} \to \mathbb{R}^{n_j}$ can be shared among some or all of the agents. For this work, it is also assumed that the agents have no coupled dynamics, but the above formulation could easily be extended to cover that case too.

The individual motion plans in this work are specified by navigating from an initial state $x_{\text{init}}^{(j)}$ to a final desired state $x_{\text{des}}^{(j)}$, within a given planning horizon $N \in \mathbb{N}$. The overall performance of the motion plan is determined by the minimization of the convex objective function $J^{(j)}(x_k^{(j)}, u_k^{(j)})$.

Additionally, certain limitations on the kinematics or actuation of the agents may be imposed to closer reflect some physical properties of the modeled system. These restrictions can be captured by using the vector-valued inequality $g^{(j)}(x_k^{(j)}, u_k^{(j)}) \preceq 0$.

Since the problem assumes centralized control, all individual vectors and constraint functions are stacked to produce the global problem formulation as given by (1) to (4).

Finally, this work considers the additional constraint of collision avoidance of the agents; which is formulated as (5) where, $\mathcal{X}(x)$ is the physical space occupied by an agent x. The condition $j > i$ esnures that for each agent-pair, only one constraint is considered. The challenge lies in incorporating the last constraint of (5), in a problem formulation accepted by a general non-linear solver.

3 Minimum-set-distance Collision Avoidance

Let $\mathcal{X}, \mathcal{Y}$ be two polygons, written as

$$\mathcal{X} = \{x \mid A\,x \preceq b\}, \quad \mathcal{Y} = \{y \mid G\,y \preceq h\}. \tag{6}$$

The shortest distance between the two polygons is given by

$$\min_{x,y} \|x - y\|, \quad \text{subject to } \begin{cases} A\,x \preceq b, \\ G\,y \preceq h. \end{cases} \tag{7}$$

The task becomes a matter of minimizing the distance between any two points belonging to each set. The Lagrange dual of the problem is given by:

$$\max_{\lambda,\mu} \; -\lambda^\top b - \mu^\top h, \quad \text{subject to } \begin{cases} \lambda^\top A + \mu^\top G = 0, \\ \|\lambda^\top A\|_* \leq 1, \\ \lambda \succeq 0,\ \mu \succeq 0, \end{cases} \tag{8}$$

where $\|\cdot\|_*$ is the dual norm of the primal problem. And so, if there exist dual vectors λ and μ which satisfy the constraints, we can guarantee a minimum distance d_{min} between the polygons, by further constraining the objective function.

Full-dimensional Agents

For this section, we assume that the full-dimensional space occupied by each agent can be defined as

$$\mathcal{X}(x^{(j)}) = \left\{\xi \mid A^{(j)}\,R(x^{(j)})\,\big(\xi - t(x^{(j)})\big) \preceq b^{(j)}\right\} \tag{9}$$

which describes the transformation of some initial pose $\mathcal{X}_0$, by rotation matrix $R : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ and translation vector $t : \mathbb{R}^n \to \mathbb{R}^n$.

In order to satisfy the collision avoidance constraint (5) using the dual reformulation, we can define the set of collision pairs;

$$\mathcal{C} = \big\{(i,j) \in \mathcal{M} \times \mathcal{M} \mid j > i\big\}. \tag{10}$$

For two agents, we get only one pair $\mathcal{C} = \{(1,2)\}$, and this number increases at a triangular rate, $|\mathcal{C}| = \frac{M^2 - M}{2}$. Using this notion of collision pairs, we write the following formulation.

$$\begin{aligned}
\min_{\boldsymbol{u},\boldsymbol{\lambda}} \; & \sum_{k=1}^{N} J(x_k, u_k), \\
\text{s.t. } & x_0 = x_{\text{init}},\ x_N = x_{\text{des}} \\
& x_{k+1} = f(x_k, u_k), \\
& g(x_k, u_k) \preceq 0, \\
& -\lambda_k^{(i)\top}(A^{(i)}\,R(x_k^{(i)})\,t(x_k^{(i)}) + b^{(i)}) + \\
& -\lambda_k^{(j)\top}(A^{(j)}\,R(x_k^{(j)})\,t(x_k^{(j)}) + b^{(j)}) \succeq d_{\text{min}} && (11) \\
& \lambda_k^{(i)\top} A^{(j)}\,R(x_k^{(i)}) + \lambda_k^{(j)\top} A^{(j)}\,R(x_k^{(j)}) = 0 && (12) \\
& \|\lambda_k^{(i)\top} A^{(i)}\,R(x_k^{(i)})\|_* \leq 1, && (13) \\
& \lambda_k^{(i)} \succeq 0,\ \lambda_k^{(j)} \succeq 0, && (14)
\end{aligned}$$

for $k = 1, \ldots, N$, and each agent-pair $(i,j) \in \mathcal{C}$.

The new problem formulation is an extension of the more general planning problem given by Eqs. (1) to (4). The additional constraints (11), (13) and (14) come from incorporating the dual of the constrained projection problem into the convex planning problem. Constraint (13) is convex regardless of the choice of norm, and (14) is affine, thus convex. However, since (11) is bilinear in terms of x and λ, the optimization problem is no longer convex.

Point-mass Agents and Proximity Polytopes

While using a full-dimensional model allows for very high precision of the collision avoidance, it does also result in a large optimization program, and the computation takes a considerable amount of time to complete. One common way to simplify the collision avoidance problem is to approximate the agents as point-mass objects, and to force the agents to always maintain a minimum separation. By choosing the minimum distance appropriately, it can serve as a conservative over-approximation of the full-dimensional problem.

To realize this using the minimum-set-distance method, we introduce the notion of *proximity polytopes*, which serve as indicators of the agents' proximity to each other. For this case, the derivation is a special case of the previous method, where one of the points is fixed in space. Consider the polytope $\mathcal{X} = \{x \mid A\,x \preceq b\}$. The dual of the minimum distance between the point $y \notin \mathcal{X}$, and the set $\mathcal{X}$ is given by

$$\max_{\lambda} \lambda^\top (A\,y - b), \quad \text{subject to } \begin{cases} \|\lambda^\top A\|_* \leq 1, \\ \lambda \succeq 0, \end{cases} \tag{15}$$

which is a simplified version of the full-dimensional case. For the proof, see (Boyd and Vandenberghe 2004, p.401). Omitting the rotation, i.e. letting $R(x) \equiv I_n$, and using the stacked state vector, we introduce one proximity polytope

$$\mathcal{X}^{(i,j)} = \left\{ \xi \mid A^{(i,j)} \xi \preceq b^{(i,j)} \right\} \tag{16}$$

and dual vector $\lambda^{(i,j)}$, for each agent pair $(i, j) \in \mathcal{C}$.

Putting it all together, we get

$$\begin{aligned} \underset{\boldsymbol{u},\boldsymbol{\lambda}}{\text{minimize}} \quad & \sum_{k=1}^{N} J(x_k, u_k), \\ \text{subject to } & x_0 = x_{\text{init}},\ x_N = x_{\text{des}} \\ & x_{k+1} = f(x_k, u_k), \\ & g(x_k, u_k) \preceq 0, \\ & (A^{(i,j)}\,x_k - b^{(i,j)})^\top \lambda_k^{(i,j)} \geq 0 \qquad (17) \\ & \|A^{(i,j)\top} \lambda_k^{(i,j)}\|_* \leq 1, \qquad (18) \\ & \lambda_k^{(i,j)} \succeq 0 \qquad (19) \end{aligned}$$

for $k = 1, \ldots, N$, and each agent-pair $(i, j) \in \mathcal{C}$.

4 Numerical Evaluation

The proposed framework was evaluated by constructing a simple case where a collision will be inevitable if no collision avoidance measures are taken. The scenario consists of two identical agents ($M = 2$) that have to reach opposing ends of a square workspace, while spending as little energy as possible. The two-dimensional state-space $\mathcal{S}$ is given by

$$\mathcal{S} = \{(x, y) \mid (0 \leq x \leq 5) \wedge (0 \leq y \leq 5)\}.$$

To increase the performance of the optimizer and guarantee that there is a feasible solution, the solver was warm-started with an intial, suboptimal trajectory, where each agent moved along the edges of the state space.

Full-dimensional Agents

For the full-dimensional case, the agents are modeled using a continuous-time kinematic bicycle model,

$$\dot{x} = v \cdot \cos\varphi, \qquad \dot{y} = v \cdot \sin\varphi, \qquad \dot{\varphi} = v \cdot \tan\gamma,$$

where (x, y) gives the position of the rear axle, φ denotes the yaw angle relative to the x-axis. The model assumes control over the velocity v and steering angle γ. The discrete-time dynamic model is elicited using an explicit Euler approximation with step-size $T > 0$, and the state and input vectors are chosen as follows

$$x_{k+1} = \begin{bmatrix} x_k(1) + u_k(1) \cdot T \cdot \cos x_k(3) \\ x_k(2) + u_k(1) \cdot T \cdot \sin x_k(3) \\ x_k(3) + u_k(1) \cdot T \cdot \tan u_k(2) \end{bmatrix}. \tag{20}$$

The agents $j = 1, 2$ are modeled as rectangles, given by

$$A^{(j)} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}^\top, \quad b^{(j)} = \begin{bmatrix} \frac{3}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}^\top$$

and geometric transformations are defined as

$$R(x) = \begin{bmatrix} \cos\big(x(3)\big) & -\sin\big(x(3)\big) \\ \sin\big(x(3)\big) & \cos\big(x(3)\big) \end{bmatrix}, \quad t(x) = \begin{bmatrix} x(1) \\ x(2) \end{bmatrix}.$$

The minimum distance was chosen as $d_{\text{min}} = 0$, and the energy cost for each agent was calculated as the absolute value of the velocity, $|u_k^{(j)}(1)|$.

Point-mass Agents – Performance Comparison

The agents were modeled as simple integrators, $x_{k+1} = x_k + T\,u_k$, where the state vector $x_k^{(j)} \in \mathbb{R}^2$ represents the coordinates of each agent in the 2D-space, and direct control of the velocity $u_k^{(j)} \in \mathbb{R}^2$ is assumed.

To formulate the example as a linear problem, we consider constraints on the ℓ_1-norm of the input. In this particular case, we consider the case where $\|u_k\|_1 \leq 1$, for each step k. The energy cost for each agent was calculated as the Euclidian norm of the velocity, $\|u_k^{(j)}\|_2^2$.

For comparison, the formulation used in (Burger and Lauer 2018) is used, namely the *Big-M* reformulation. Instead of the real-valued λ_k, this method incorporates the binary (zero or one) decision-vector at each step, δ_k, which means that the problem becomes a mixed-integer program. By retaining the notion of the proximity polytope, we can replace constraints (17), (18) and (19) with

$$A^{(i,j)}\,x_k - b^{(i,j)} \succeq (\delta_k^{(i,j)} - \mathbf{1}) \cdot M_\delta \tag{21}$$

$$\mathbf{1}^\top \delta_k^{(i,j)} \geq 1, \tag{22}$$

where M_δ is a sufficiently-large design scalar which gives the method its name. In this case, we choose $M_\delta = 2x_{\text{max}}$.

The performance comparison was done by running the test scenario with increasing planning horizons, and proportionally decreasing time-steps T. For the modelling, YALMIP (Löfberg 2004) was used, and the primal-dual interior-point solver IPOPT (Wächter and Biegler 2006) was used for solving the non-convex problem. The mixed-integer program was solved using the commercial optimizer Gurobi (Gurobi Optimization, LLC 2021). Both were warm-started in the same way.

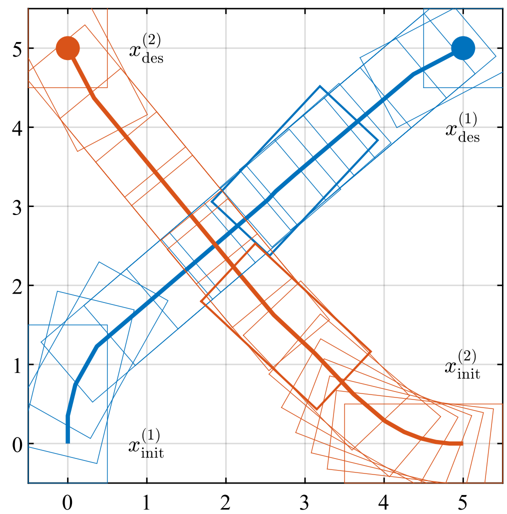
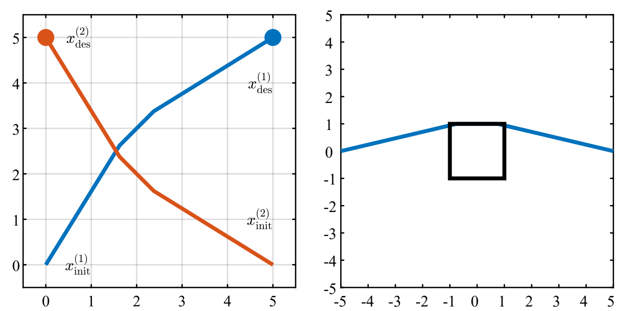


Figure 1: Collision avoidance for full-dimensional, multi-agent trajectory optimization. The slightly thicker contours shows the agents' pose when they are at their closest.



(a) State-space representation (b) Agent proximity vector

Figure 2: Two-dimensional test scenario with two simple integrator agents and proximity polytope collision avoidance

5 Results

The results for the full-dimensional trajectory optimization can be seen in Figure 1. As can be seen, both agents complete their motion tasks without colliding. The overall behavior consists of agent 1 speeding up, and agent 2 slowing down so that they can clear each other's path. Since the minimum distance is chosen to be zero, the planner converges on a trajectory where the two agents pass each other with no extra safety margin. This can also be seen in the figure, where the seventh time-step is highlighted, revealing that the agents turn slightly to avoid the occupancy sets intersecting.

The resulting motion plan for the linear test scenario is shown in Figure 2. As can be seen in (a), the agents both turn slightly left, resulting in the agents also clearing each other, while still keeping the minimal distance. This can be more clearly seen in (b), in that the proximity vector never penetrates the proximity polytope centered at the origin. The trajectory is indistinguishable from the the mixed-integer program output.

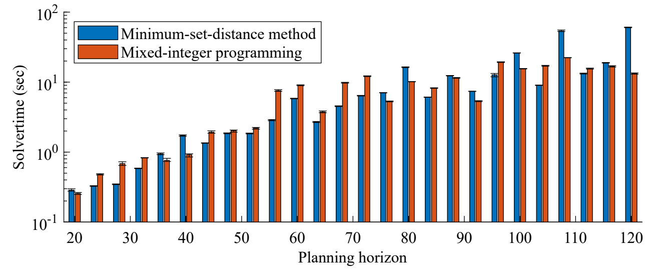


Figure 3: Comparing the solvertime of the new method against mixed-integer programming

Figure 3 shows the solvertime for the increasingly larger optimization problems in a logarithmic scale. As can be seen, the proposed solution performs comparably well. For the small-to-medium sized problems, it can be seen that the new method performs slightly better on average.

6 Conclusion

To summarize, the minimum-set-distance method has been demonstrated as suitable for trajectory generation for multi-agent systems with full-dimensional occupancy, and non-linear dynamics. Additionally, when simplifying the dynamics to be linear, the proposed planning framework produces an equivalent plan for the test scenario, when compared to the mixed-integer programming approach; avoiding the collision, while still staying suffiently close as to minimize the energy spent.

Lastly, when evaluating the computational performance, it can be seen that the method has an equivalent performance, when comparing solvertime against the MIP approach on the linear test scenario, and making the same approximations.

7 Discussion & Future Work

One of the most important aspect of the proposed approach is that it can be used for general non-linear dynamical systems, such as differential-drive robots, bicycle models, or other common kinematic models. Additionally, it is possible to reach high precision, and representing the objects as full-dimensional convex sets, avoiding further simplifications. However, it should be reiterated that in order to guarantee a feasible solution, the solver has to be warm-started by providing an initial, sub-optimal "guess". Otherwise, the solver runs the risk of converging to locally infeasible points.

Finally, it should also be mentioned that owing to the non-convex nature of the optimization problem, the solver will only converge to a local minimum, as determined by the inital guess. Some investigation into what constitutes a valid initial guess was conducted. For example, providing only a sequence of poses without the corresponding inputs was sufficient for the solver to converge. For future work, this will be examined further.

Acknowledgments

This section is left intentionally blank for blinding purposes.

References

Alrifaee, B.; Kostyszyn, K.; and Abel, D. 2016. Model Predictive Control for Collision Avoidance of Networked Vehicles Using Lagrangian Relaxation. *IFAC-PapersOnLine*, 49(3): 430–435.

Andreychuk, A.; Yakovlev, K.; Atzmon, D.; and Stern, R. 2019. Multi-Agent Pathfinding with Continuous Time. *arXiv:1901.05506 [cs]*.

Boyd, S. P.; and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press. ISBN 978-0-521-83378-3.

Burger, C.; and Lauer, M. 2018. Cooperative Multiple Vehicle Trajectory Planning Using MIQP. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 602–607.

Goerzen, C.; Kong, Z.; and Mettler, B. 2010. A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4): 65–100.

Gurobi Optimization, LLC. 2021. Gurobi Optimizer Reference Manual. https://www.gurobi.com.

Li, S. E.; Wang, Z.; Zheng, Y.; Sun, Q.; Gao, J.; Ma, F.; and Li, K. 2020. Synchronous and Asynchronous Parallel Computation for Large-Scale Optimal Control of Connected Vehicles. *Transportation Research Part C: Emerging Technologies*, 121: 102842.

Liu, Z.; Wu, B.; Dai, J.; and Lin, H. 2020. Distributed Communication-Aware Motion Planning for Networked Mobile Robots under Formal Specifications. *IEEE Transactions on Control of Network Systems*, 7(4): 1801–1811.

Löfberg, J. 2004. YALMIP: A Toolbox for Modeling and Optimization in MATLAB. In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 284–289.

Mohanan, M.; and Salgoankar, A. 2018. A Survey of Robotic Motion Planning in Dynamic Environments. *Robotics and Autonomous Systems*, 100: 171–185.

Ong, H. Y.; and Gerdes, J. C. 2015. Cooperative Collision Avoidance via Proximal Message Passing. In *2015 American Control Conference (ACC)*, 4124–4130. Chicago, IL, USA: IEEE. ISBN 978-1-4799-8684-2.

Paden, B.; Čáp, M.; Yong, S. Z.; Yershov, D.; and Frazzoli, E. 2016. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1): 33–55.

Prodan, I.; Olaru, S.; Stoica, C.; and Niculescu, S.-I. 2010. Collision Avoidance and Path Following for Multi-Agent Dynamical Systems. In *ICCAS 2010*, 1930–1935.

Soloperto, R.; Kohler, J.; Allguwer, F.; and Muller, M. A. 2019. Collision Avoidance for Uncertain Nonlinear Systems with Moving Obstacles Using Robust Model Predictive Control. In *2019 18th European Control Conference (ECC)*, 811–817. Naples, Italy: IEEE. ISBN 978-3-907144-00-8.

Stern, R.; Sturtevant, N.; Felner, A.; Koenig, S.; Ma, H.; Walker, T.; Li, J.; Atzmon, D.; Cohen, L.; Satish Kumar, T.; Boyarski, E.; and Barták, R. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. In *Proceedings of the 12th International Symposium on Combinatorial Search, SoCS 2019*, 151–158. ISBN 978-1-57735-808-4.

Wächter, A.; and Biegler, L. T. 2006. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1): 25–57.

Zhang, X.; Liniger, A.; and Borrelli, F. 2021. Optimization-Based Collision Avoidance. *IEEE Transactions on Control Systems Technology*, 29(3): 972–983.