

RADMPCC: A Fast Decentralized Approach for Chance-Constrained Multi-Vehicle Path-Planning

Aaron Huang, Benjamin J. Ayton, Brian C. Williams

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
32 Vassar Street
Cambridge, MA 02139

Abstract

Robust multi-vehicle path-planning is important for ensuring the safety of multi-vehicle systems in applications like transportation, search and rescue, and robotic exploration. Chance-constrained methods like Iterative Risk Allocation (IRA)(Ono and Williams 2008) have been developed for situations where environmental disturbances are unbounded. However, chance-constrained methods for the multi-vehicle case generally use centralized strategies where the vehicle set is planned with couplings between all vehicle pairs. This approach is intractable as fleet size increases because computation time is exponential with respect to the number of vehicles being planned over due to a polynomial increase in coupling constraints between vehicle pairs. We present a faster approach for chance-constrained multi-vehicle path-planning that relies upon a decentralized path-planning method called Risk-Aware Decentralized Model Predictive Control (RADMPCC) to rapidly approximate a centralized IRA approach. The RADMPCC approximation is evaluated for vehicle interactions to determine the vehicle sets that should be planned in a coupled manner. Applying IRA to the smaller vehicle sets determined from the RADMPCC approximation rapidly plans safe paths for the entire fleet. A Monte Carlo simulation analysis demonstrates the correctness of our approach and a significant improvement in computation time compared to a centralized IRA approach.

Introduction

Recent interest in the utilization of autonomous vehicle fleets has skyrocketed for applications like urban transportation and undersea exploration. There is an imminent need for architectures that support safe coordination of multiple vehicles in a practical and optimal manner. A critical element of multi-vehicle coordination is safe multi-vehicle path-planning in an environment where collisions are possible. Paths for autonomous vehicle fleets should be optimal by some measure (e.g. actuation cost, solution execution time) and must have a reasonable computation time to be practical.

Prior methods for fast multi-vehicle path-planning are primarily fixated on situations where environmental disturbances are assumed to be bounded. Robust Model Predictive Control (RMPC) techniques have been extensively stud-

ied for multi-vehicle path-planning in this case. However, the assumption of bounded environmental disturbances is invalid in many practical applications. Chance-constrained techniques are commonly used for path-planning in the alternate case where environmental disturbances are assumed to be unbounded. Current chance-constrained techniques for the multi-vehicle case generally use a *centralized* strategy where the couplings are present for all vehicle pairs in the entire vehicle set. Centralized strategies are intractable as fleet size increases because computation time is typically exponential as more vehicles are considered. This is due to a combinatorial increase in binary variables that are required for vehicle and obstacle avoidance constraints.

We present a faster approach to chance-constrained multi-vehicle path-planning that relies upon a novel method called Risk-Aware Decentralized Model Predictive Control (RADMPCC). RADMPCC extends from Decentralized Model Predictive Control (DMPC) techniques by using a single iteration of IRA as the primary optimization method. RADMPCC plays a key role in reducing the complexity of the multi-vehicle path-planning problem by rapidly approximating a centralized application of IRA. The approximation is examined for vehicle interactions to determine smaller sets of vehicles that need to be planned in a coupled manner. Individually applying IRA to the smaller vehicle sets plans safe paths for the vehicle fleet much more quickly than a centralized IRA approach.

Literature Review

There is a considerable body of work in path-planning for multiple autonomous vehicles. This is a hard problem because of two reasons: the presence of environmental uncertainty and the nonconvexity of the optimization problem. Mixed-Integer Linear Programming (Schouwenaars et al. 2001) and Disjunctive Linear Programming (Balas 1998) strategies were used in initial techniques to handle the non-convex optimization problems. However, these approaches did not account for sources of uncertainty in the problem.

Many methods that handle uncertainty usually make the assumption of bounded uncertainties and environmental disturbances, allowing for the design of robust trajectories that are resistant to constraint failure against the worst case disturbances. Robust Model Predictive Control (RMPC) is an extension of Model Prediction Control (MPC) that is com-

monly used for path-planning under this assumption. Trajectories generated with RMPC techniques have been shown to be safe from obstacle collision at up to 3σ confidence at each timestep (Pepy and Lambert 2006; Alexis et al. 2015; Jalali and Nadimi 2006). However, RMPC techniques are still intractable for multi-vehicle problems with larger groups of vehicles.

Decentralized Model Predictive Control (DMPC) (Richards and How 2004) algorithms employ the strategy of decomposing the full multi-vehicle trajectory optimization problem into decentralized subproblems to reduce computational intensity. Each subproblem optimizes the trajectory of a single vehicle using an RMPC strategy. To account for inter-vehicle coupling constraints, each subproblem must be solved while considering all other vehicle trajectories. DMPC techniques are computationally inexpensive compared to RMPC techniques for multi-vehicle problems.

Many real-life situations involve uncertainties that cannot be bounded, making it impossible to guarantee constraint satisfaction with zero probability of failure using most RMPC and DMPC approaches. In lieu of guaranteed constraint satisfaction, a different approach to path-planning is to place *chance constraints* to limit the *probability* of violating constraints. There are two kinds of chance constraints: *individual* chance constraints limiting the probability of failure of a single constraint at a single timestep, and *joint* chance constraints limiting the probability of failure of any constraint in a problem (Li, Wendt, and Wozny 2002).

Solving a multi-vehicle path-planning problem with a joint chance constraint is hard because it requires the computationally intensive calculation of the probabilities of non-independent events. Blackmore, Ono, and Williams demonstrate an elegant method in which Boole's inequality is used to decompose the joint chance constraint into individual chance constraints. The new RMPC problem with individual constraints can be more easily solved by constraint tightening. However, this chance-constrained method is conservative since it assigns a uniform value to the risk bound for every individual chance constraint.

Iterative Risk Allocation (IRA) (Ono and Williams 2008) is a two-stage optimization method that uses the concept of *risk reallocation* to reduce the suboptimality of chance-constrained approaches. By iteratively reallocating risk from *inactive* chance constraints to *active* chance constraints, the allocation of risk that optimizes the objective function can be found. Unfortunately, the centralized application of IRA still has an exponential solution time and is impractical for problems with many vehicles.

A strategy for reducing the complexity of centralized approaches is to decouple unnecessary coupling constraints between vehicle pairs using heuristics. Keviczky et al. uses a distance-based heuristic for multi-vehicle path-planning to maintain a communication topology graph that is updated over time. Undirected edges between any two vehicles represent a coupling constraint indicating that either vehicle must account for the other vehicle's actions when planning. However, the distance-based heuristic does not make full use of the vehicle information available when determining coupling constraints.

Problem Statement

Notation

The following notation is used throughout the paper

- \mathbf{x}_i^k : State vector for vehicle i at time k
- \mathbf{u}_i^k : Control input for vehicle i at time k
- \mathbf{w}_i^k : Disturbance for vehicle i at time k
- $\bar{\mathbf{x}}_i^k := E[\mathbf{x}_i^k]$: Nominal state for vehicle i at time k
- δ_j^k : Risk bound for chance constraint j at time k
- Δ : Risk bound for the joint chance constraint

$$\mathbf{X} := \begin{bmatrix} \mathbf{x}_0^0 \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} \quad \bar{\mathbf{X}} := \begin{bmatrix} \bar{\mathbf{x}}_0^0 \\ \vdots \\ \bar{\mathbf{x}}_N^T \end{bmatrix} \quad \mathbf{U} := \begin{bmatrix} \mathbf{u}_0^0 \\ \vdots \\ \mathbf{u}_N^T \end{bmatrix}$$

$$i \in [0 \dots N] \quad j \in [0 \dots L] \quad k \in [0 \dots T]$$

N denotes the number of vehicles to plan over. T denotes the total number of timesteps in the problem. L refers to the number of chance constraints present.

RMPC with a joint chance constraint

The chance-constrained multi-vehicle path-planning problem is formulated as follows:

$$\min_{\mathbf{U}} E[J(\mathbf{X}, \mathbf{U})] \quad (1)$$

$$\text{s.t.} \quad \mathbf{x}_i^{k+1} = A\mathbf{x}_i^k + B\mathbf{u}_i^k + \mathbf{w}_i^k \quad (2)$$

$$\mathbf{u}_{i,\min} \leq \mathbf{u}_i^k \leq \mathbf{u}_{i,\max} \quad (3)$$

$$\mathbf{w}_i^k \sim \mathcal{N}(0, \Sigma_{\mathbf{w}_i^0}) \quad (4)$$

$$\mathbf{x}_i^0 \sim \mathcal{N}(\bar{\mathbf{x}}_i^0, \Sigma_{\mathbf{x}_i^0}) \quad (5)$$

$$\Pr \left[\bigwedge_{i=0}^N \bigwedge_{j=0}^L \bigwedge_{k=0}^T \mathbf{h}_j^{kT} \mathbf{x}_i^k \leq g_j^k \right] \geq 1 - \Delta \quad (6)$$

We model vehicles as discrete-time linear time invariant (LTI) systems operating in the presence of *unbounded* environmental disturbances. In Equation 1, we wish to determine the control sequence \mathbf{U} that generates a state sequence \mathbf{X} that minimizes the expected value of the objective function J while obeying Equations 2 - 6. Equation 2 defines state evolution for vehicles from time k to time $k + 1$ where matrices A and B represent linear vehicle dynamics and control effects. The combination of Equation 2 and Equation 4 explicitly describe the effects of an unbounded Gaussian disturbance \mathbf{w}_i^k , forcing vehicle states to be unbounded. Equation 6 defines the total risk bound Δ as the upper bound on the probability that any individual chance constraint fails.

Preliminaries

We will briefly describe IRA and the decomposition of the joint chance constraint into individual chance constraints to provide better context for our approach.

RMPC with individual chance constraints

We cannot easily solve RMPC with a joint chance constraint because Equation 6 involves the integration of a multivariate Gaussian distribution. However, this problem can be made more tractable by decomposing the joint chance constraint into individual chance constraints using Boole's inequality ($\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$).

$$\begin{aligned} \min_{\mathcal{U}} \quad & E[J(\mathbf{X}, \mathbf{U})] \\ \text{s.t.} \quad & \mathbf{x}_i^{k+1} = A\mathbf{x}_i^k + B\mathbf{u}_i^k + \mathbf{w}_i^k \\ & \mathbf{u}_{i,\min} \leq \mathbf{u}_i^k \leq \mathbf{u}_{i,\max} \\ & \mathbf{w}_i^k \sim \mathcal{N}(0, \Sigma_{\mathbf{w}_i^0}) \\ & \mathbf{x}_i^0 \sim \mathcal{N}(\bar{\mathbf{x}}_i^0, \Sigma_{\mathbf{x}_i^0}) \\ \forall i, j, k \quad & \Pr[\mathbf{h}_j^{kT} \mathbf{x}_i^k \leq g_j^k] \geq 1 - \delta_j^k \quad (7) \\ & \sum_{j=0}^L \sum_{k=0}^T \delta_j^k \leq \Delta \quad (8) \end{aligned}$$

Decomposing the joint chance constraint directly reduces Equation 6 to Equation 7, with Equation 8 constraining the sum of all δ_j^k to be no greater than Δ . Blackmore, Ono, and Williams demonstrate a method to solve this problem that transforms the stochastic problem into a deterministic problem. However, their method produces a suboptimal solution since it fixes all δ_j^k using a uniform allocation of Δ . In other words, there may be an allocation of δ_j^k that produces a more optimal solution.

Iterative Risk Allocation

IRA is a method that is designed to achieve greater solution optimality for RMPC problems with joint chance constraints. IRA is a two-stage optimization method that uses *risk reallocation* to determine the best risk allocation of δ_j^k that optimizes the objective function. The innovation of risk reallocation is in moving risk from inactive constraints to active constraints to monotonically decrease overall cost.

To determine if a constraint is active or inactive, we must compare an individual chance constraint's risk bound to the probability of constraint satisfaction. This requires the probability of constraint satisfaction to be reformulated using deterministic variables. Recall that an individual chance constraint is defined as follows:

$$P(\mathbf{h}_j^{kT} \mathbf{x}_i^k \leq g_j^k) \geq 1 - \delta_j^k$$

The lower stage optimization in IRA computes a solution for \mathbf{x}_i^k using the fixed δ_j^k for each constraint. Then, we can define a constraint for the acceptable values of δ_j^k by writing the probability of constraint satisfaction in terms of a deterministic cumulative distribution function of \mathbf{x}_i^k .

$$\begin{aligned} P(\mathbf{h}_j^{kT} \mathbf{x}_i^k \leq g_j^k) &= \text{cdf}(g_j^k - \mathbf{h}_j^{kT} \mathbf{x}_i^k) \\ \implies \delta_j^k &\geq 1 - \text{cdf}(g_j^k - \mathbf{h}_j^{kT} \mathbf{x}_i^k) \end{aligned}$$

This result defines the minimum value for δ_j^k that is required to satisfy the chance constraint given the current solution for \mathbf{x}_i^k . Active and inactive constraints are determined by comparing the fixed δ_j^k to the newly defined $\delta_{j,\min}^k$ using a tolerance η .

$$\begin{aligned} \delta_{j,\min}^k &= 1 - \text{cdf}(g_j^k - \mathbf{h}_j^{kT} \mathbf{x}_i^k) \\ \text{Active:} \quad & |\delta_j^k - \delta_{j,\min}^k| \leq \eta \\ \text{Inactive:} \quad & |\delta_j^k - \delta_{j,\min}^k| > \eta \end{aligned}$$

The risk reallocation upper stage reallocates risk from active to inactive constraints while respecting the minimum risk bound. This defines a new risk allocation to be used in the next iteration of the lower optimization stage. This two-stage procedure is iteratively run for a fixed number of iterations or until the value of the objective function converges. Although this method produces excellent results, a centralized application of IRA to large multi-vehicle problems is computationally intensive because the number of avoidance constraints increases combinatorially.

Avoidance Constraints

Vehicle and obstacle avoidance constraints are defined by a disjunction of individual chance constraints each represented as a linear inequality. For a set of individual chance constraints that comprise an object O to be avoided, the following disjunction of inequalities define a safe zone.

$$\bigvee_j^O \mathbf{h}_j^{kT} \mathbf{x}_i^k \leq g_j^k$$

However, using a disjunction of linear inequalities for obstacle avoidance is insufficient for path-planning because a vehicle cannot simultaneously be on every side of an object at once. Instead, we use a conjunction of the same linear inequalities and introduce binary variables b_k^j that are multiplied with an arbitrarily large number M to turn off boundary inequalities as needed. To ensure that at least one constraint is turned on, the sum of all binary variables must be less than the number of individual chance constraints that comprise O .

$$\begin{aligned} \bigwedge_j^O \mathbf{h}_j^{kT} \mathbf{x}_i^k &\leq g_j^k + Mb_j^k \\ \sum_j^O b_j^k &\leq \text{card}(O) - 1 \end{aligned}$$

This formulation induces a polynomial increase in binary variables as more objects are considered. Vehicle avoidance constraints are similarly encoded. However, computation time is more adversely affected by additional vehicles than additional obstacles. This follows because avoidance constraints must exist between every vehicle pair, which grows combinatorially as more vehicles are added. A problem with N vehicles must include $\binom{N}{2}$ vehicle avoidance

constraints over T timesteps. This implies that binary variables for vehicles increase with $O(N^2T)$ complexity and force an $O(e^{N^2T})$ solution time. On the other hand, additional obstacles increase binary variables with $O(NT)$ complexity, adding only $O(e^{NT})$ complexity. This distinction is important as we will later use a special form of obstacle called a *temporal obstacle* to represent vehicles, reducing RADMPCC complexity.

Technical Approach

We propose a three-step approach to produce the result of a centralized application of IRA to a large vehicle set with much less computational overhead. We present RADMPCC - a fast, risk-aware path-planner used to approximate centralized IRA. The RADMPCC approximation is evaluated to identify vehicle pairs with high probability of collision and decompose the full set of vehicles into smaller vehicle subsets that have only *relevant* vehicle coupling constraints. Finally, we apply IRA to the smaller subsets to produce paths for all vehicles far more quickly than centralized IRA. Figure 1 depicts the flow of our approach. Figure 2 depicts example plots of different stages of our approach. The runtime of the centralized IRA solution exceeds the combined runtime of the RADMPCC approximation and the runtime of applying IRA to the two vehicle subsets.

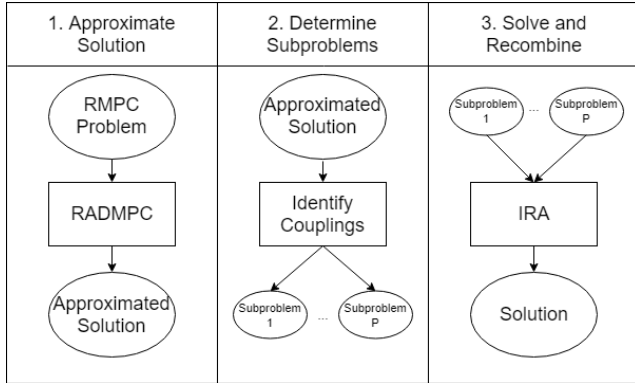


Figure 1: The three stages of our approach

Algorithm 1 Multi-Vehicle

```

1: procedure MULTI-VEHICLE( $V, N, T, \Delta, \Psi$ )
2:    $\mathbf{X}_{appr} \leftarrow \text{RADMPCC}(V, N, T, \Delta)$ 
3:    $V^* \leftarrow \text{findCouplings}(\mathbf{X}_{appr}, \Psi)$ 
4:   for each coupled set  $v^*$  in  $V^*$  do
5:      $\Delta^* \leftarrow \Delta \cdot \text{card}(v^*)/N$ 
6:      $\text{IRA}(v^*, N, T, \Delta^*)$ 
7:   end for
8: end procedure

```

RADMPCC

We have developed a novel path-planning algorithm called Risk Aware Decentralized Model Predictive Control

(RADMPCC) to be used as the fast path-planner in our approach. To be useful, it must be able to closely approximate centralized IRA. To be practical, it cannot be computationally expensive lest we fail to handle the core issue of solution time complexity for multi-vehicle path-planning. RADMPCC is an extension of Decentralized Model Predictive Control (DMPC) as proposed by Richards and How. However, our approach differs from DMPC by assuming unbounded environmental disturbances as opposed to bounded disturbances and using *temporal obstacles* to communicate vehicle plans across subproblems.

RADMPCC uses a decentralized path-planning strategy to quickly plan paths. Given N vehicles, RADMPCC decomposes the full problem into N subproblems that each optimize the trajectory of a single vehicle. At time k , the subproblems are solved in a randomized order. The optimization method used to solve each subproblem simply consists of a single iteration of IRA with a uniform initial allocation over a *risk pool*. After a subproblem is solved, the first control input of the solution is executed and a temporal obstacle is created to bound the vehicle's subproblem solution. RADMPCC is recursively executed in this way until all vehicles are in their respective goal.

Algorithm 2 RADMPCC

```

1: procedure RADMPCC( $V, N, T, \Delta$ )
2:    $O \leftarrow \text{initTemporalObstacles}(V)$ 
3:    $\Delta^\# \leftarrow \Delta, N^* \leftarrow N$ 
4:   for  $k$  in  $[0 \dots T - 1]$  do
5:     for  $i$  in random ordering of  $[0 \dots N - 1]$  do
6:        $\mathbf{X}_i^*, \delta_i^* \leftarrow \text{fastIRA}(V_i, T - k, \Delta^\# / N^*)$ 
7:        $\mathbf{x}_{i,appr}^{k+1} \leftarrow \mathbf{x}_i^{*1}$ 
8:       if all vehicles in goal then
9:         return  $\mathbf{X}_{appr}$ 
10:      else if  $V_i$  in goal then
11:         $N^* \leftarrow N^* - 1$ 
12:      end if
13:       $O_i \leftarrow \text{makeTemporalObstacle}(\mathbf{X}_i^*)$ 
14:       $\Delta^\# \leftarrow \Delta^\# - \text{sum}(\delta_i^{*1})$ 
15:    end for
16:  end for
17: end procedure

```

Risk Pooling A *risk pool* $\Delta^\#$ is used to track the total risk remaining while executing RADMPCC. $\Delta^\#$ is initialized to Δ . Every subproblem is given a uniform risk allocation over $\Delta^\# / N^*$ where N^* is the number of vehicles that are not in goal. After a subproblem is solved, $\Delta^\#$ is updated as follows:

$$\Delta^\# \leftarrow \Delta^\# - \sum_{j=0}^L \delta_0^j$$

This allows RADMPCC to be risk-aware by subtracting the risk that is used at the *executed* timestep from the risk pool. The risk pool is then divided among the remaining vehicles for the next subproblem.

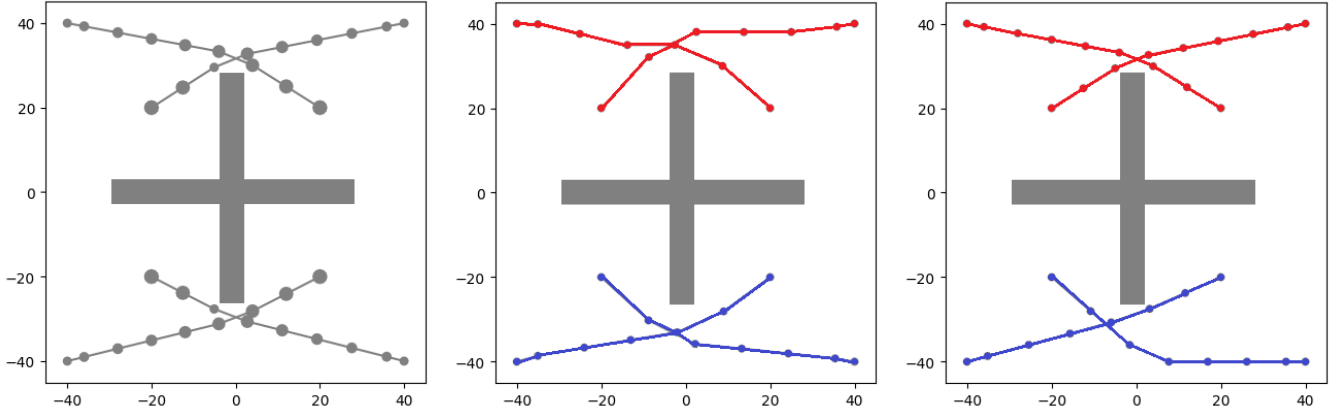


Figure 2: Left to right: centralized IRA solution, RADMPc approximation, RADMPc solution. Runtime of the RADMPc approximation and IRA on coupled vehicle sets is generally much faster than runtime of centralized IRA.

Temporal Obstacles We introduce a *temporal obstacle* representation for a vehicle’s plan according to its most recently solved subproblem (hereafter referred to as *temporal vehicle obstacles*). Temporal obstacles are a simple extension of static obstacles where the linear inequalities describing the obstacle boundaries change with respect to time.

Temporal vehicle obstacles are created after a subproblem is solved for a vehicle i using the subproblem solution \mathbf{X}_i^* . A temporal vehicle obstacle bounds the 3σ confidence region of \mathbf{x}_i^{*k} for all timesteps $k \in [0 \dots T]$ for vehicle i . We define distance d as the radial distance from $\bar{\mathbf{x}}_i^{*k}$ bounding the 3σ confidence region. As truly circular obstacles are difficult to represent, d is instead used to compute square coordinates around \mathbf{x}_i^{*k} . Square obstacles are used to represent vehicles in goal and vehicles at time $k = 0$. At other timesteps, the obstacle is generated by wrapping a convex hull around the square coordinates generated at \mathbf{x}_i^{*k} and \mathbf{x}_i^{*k+1} .

Using a temporal obstacle representation for vehicles offers a light, yet powerful representation when other vehicles need to consider their intent while planning. Representing vehicles as temporal obstacles has $O(e^{NT})$ computational complexity as opposed to $O(e^{N^2T})$ complexity otherwise. This formulation also offers an intuitive way of evaluating interactions between vehicles planned using RADMPc.

Identifying Couplings

Couplings are determined by evaluating the probability of collision between vehicles and temporal vehicle obstacles. For each boundary in a temporal vehicle obstacle that has an activated linear constraint (i.e. the associated binary variable is 0), the collision probability is computed as follows (Blackmore, Ono, and Williams 2011):

$$\Pr[\mathbf{h}_j^{kT} \mathbf{x}_i^k > g_j^k] = \frac{1}{2} - \frac{1}{2} \operatorname{erf} \left(\frac{g_j^k - \mathbf{h}_j^{kT} \bar{\mathbf{x}}_i^k}{\sqrt{2\mathbf{h}_j^{kT} \Sigma_{\mathbf{x}_i^k} \mathbf{h}_j^k}} \right)$$

After a RADMPc subproblem is solved, we record the collision probabilities between the subproblem vehicle and

the temporal vehicle obstacles *at the initial timestep*. We do not record the collision probabilities at other timesteps because RADMPc replans for the vehicle at the following timestep.

After RADMPc converges on a solution, the maximum probability of collision of a vehicle with another temporal vehicle obstacle provides a direct metric for the probability of collision between the vehicle pair. This is used to determine vehicle pairs that have significant interactions and must be planned together. For every vehicle pair, there are two candidate maximum collision probabilities since both vehicles avoid the other’s temporal vehicle obstacle when planning. The greater of the two probabilities is chosen as the maximum probability of collision for the vehicle pair. If the maximum probability of collision for the vehicle pair exceeds a given threshold Ψ , the vehicle pair must be planned in a coupled manner.

Distributed Solving

Determining coupled vehicle sets is analogous to determining the disconnected parts of a graph where vehicles are nodes and couplings are edges between vehicles. Path-planning problems are created for each coupled vehicle set. Applying IRA on every problem and combining the solutions computes the RADMPc solution to the original multi-vehicle problem.

Computation Time Analysis We now analyze the solution time of our two-step approach. For ease of notation, we will factor out runtime due to common factors between our approach and the centralized approach (e.g. static obstacles used in both approaches). A centralized application of IRA to a problem with N vehicles over T timesteps uses $O(N^2T)$ binary variables for vehicle avoidance constraints between vehicle pairs, causing a $O(e^{N^2T})$ computational complexity.

The solution time of RADMPc likewise depends on the number of vehicles in the problem and the number of timesteps it takes for vehicles to reach their goal in our receding horizon approach. Since each decentralized subprob-

lem uses temporal vehicle obstacles for vehicle avoidance, there are $O(NT)$ binary variables that imply $O(e^{NT})$ computation complexity per subproblem. Using the conservative assumption that RADMPCC requires all T timesteps to execute, there are N decentralized problems over T timesteps giving RADMPCC $O(NTe^{NT})$ total computational complexity.

Let us assume that analysis of the RADMPCC approximation separates the full vehicle set into $\frac{N}{P}$ coupled vehicle sets with P vehicles each. Applying centralized IRA on problems covering the coupled vehicle sets has $O(\frac{N}{P}e^{P^2T})$ computational complexity. Factoring in RADMPCC runtime indicates $O(NTe^{NT} + \frac{N}{P}e^{P^2T})$ complexity. In the optimal scenario, we would be able to completely separate the problem into N subproblems that are each individually solved with IRA for $O(NTe^{NT} + Ne^T)$ complexity. In the worst case, the complexity is $O(NTe^{NT} + e^{N^2T})$. However empirical results indicate that RADMPCC decouples most vehicle couplings and generally does not exhibit worst case performance.

Results

In this section, we demonstrate empirical results of the RADMPCC approach. We focus on analyzing the accuracy of the RADMPCC approximation, average runtime of our approach compared to centralized IRA, and correctness of the RADMPCC solution.

RADMPCC Approximations

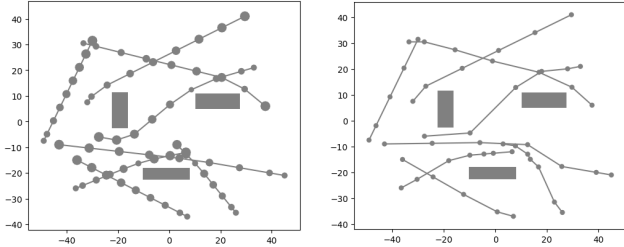


Figure 3: Centralized solution on left, RADMPCC approximation on right

Figure 3 demonstrates the RADMPCC approximation on an eight vehicle problem. The RADMPCC approximation is a good approximation of the centralized solution with only slight disturbances. A visual inspection of multiple sample problems indicates that RADMPCC generally approximates the centralized solution well enough to be useful.

However, the accuracy of the RADMPCC approximation decreases in some cases especially when the complexity of the multi-vehicle problem increases. This may be because larger vehicle sets exhibit perturbances where "ripple" effects in RADMPCC cause the displacement of a single vehicle to propagate to the other vehicles. Significant deviations from the centralized solution could lead RADMPCC to make incorrect determinations of coupled vehicle sets, adversely

affecting the correctness of the RADMPCC solution. In general, RADMPCC works the best in situations where vehicle interactions are less abundant i.e. situations where the problem should be decoupled.

Runtime Comparisons

We compare the average runtime of our approach and a centralized application of IRA by drawing 50 sample problems for each of $N \in [2, 8]$ vehicles and generating random vehicle starts and goals for each sample problem. We use a simple environment with three regular obstacle and use the following parameters:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{1}{2}dt^2 & 0 \\ 0 & \frac{1}{2}dt^2 \\ dt & 0 \\ 0 & dt \end{bmatrix}$$

$$T = 10 \quad \Delta = 0.05 \quad \Psi = 0.000001$$

$$J(\mathbf{X}, \mathbf{U}) = \sum_{i=0}^N \sum_{k=0}^T |u_k^i|$$

Figure 4 demonstrates a significant reduction in solution time when solving complex RMPC problems with the heuristic, even when RADMPCC runtime is included. As the size of the vehicle set increases, our approach will run faster by margins that increase with vehicle account. In our largest experiments involving 8 vehicles, the mean speedup factor reached 46. Much of this improvement can be attributed to the observation that most randomly generated complex problems can be separated into smaller, simpler problems with one or two vehicles each.

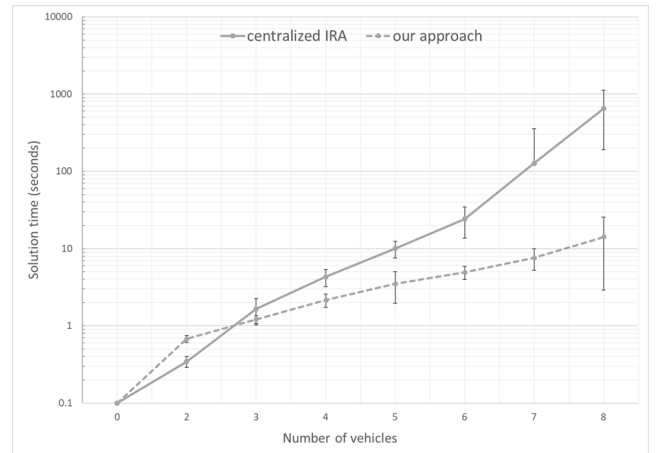


Figure 4: Runtime comparison between centralized IRA and our approach

Simulation

A Monte Carlo simulation is used to verify the correctness of the RADMPCC solution. Our Monte Carlo simulation uses

the sample problems described in the prior section. A success rate is computed for each sample problem to express correctness. A correct solution is a solution that does not exhibit a probability of collision above the risk bound. To compute vehicle collisions in the RADMPCC solution, every vehicle state x_k^i in the computed state sequence \mathbf{X} is sampled 100,000 times. Vehicle collisions are detected using a simple computation that ensures that sampled vehicle states at corresponding time k are not within two vehicle radii of each other.

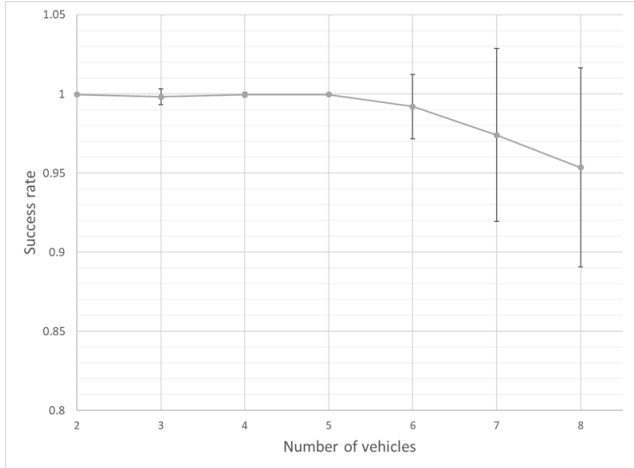


Figure 5: Success rate of our approach with different numbers of vehicles

Our approach on average works extremely well for the range of vehicles we considered for our experimentation. However, the success rate decreases as the number of vehicles increases and more deviation from the average success rate occurs. This trend can be expected to continue as the complexity of the problem increases, indicating that RADMPCC is not as well suited for approximating complex problems with extensive inter-vehicle interaction. Still, the results presented demonstrate that our approach is useful to problems that include up to eight vehicles, with possible extension to more vehicles with further development.

Conclusions

We have presented a method that intelligently decouples computationally expensive chance-constrained multi-vehicle problems to reduce solution time complexity. We describe a novel path-planning method called Risk-Aware Decentralized Model Predictive Control (RADMPCC) to determine sets of coupled vehicles that have should be planned together. Applying IRA to each set generates collision-free paths in far less time than a centralized application of IRA.

We envision our approach being used to enable more rapid planning capabilities for vehicle swarms being used in the field. Although this approach was specifically designed to support autonomous underwater vehicle exploration, fast multi-vehicle path-planning spans a large range of academic, industrial, and humanitarian applications. Fur-

ther research into fast multi-vehicle path-planning will facilitate more rapid integration of large-scale autonomous vehicle solutions in situations where they are sorely needed.

Acknowledgement

I would like to sincerely thank Benjamin Ayton for advising me on this project over the past year and helping me develop my skills as an undergraduate researcher. Thanks to Professor Brian Williams for being my faculty adviser and the MERS group for welcoming me into the family. Lastly, thanks to the MIT SuperUROP program for financially and logistically supporting this research project.

References

- [Alexis et al. 2015] Alexis, K.; Papachristos, C.; Siegwart, R.; and Tzes, A. 2015. Robust model predictive flight control of unmanned rotorcrafts. 81.
- [Balas 1998] Balas, E. 1998. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics* 89(1):3 – 44.
- [Blackmore, Ono, and Williams 2011] Blackmore, L.; Ono, M.; and Williams, B. C. 2011. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics* 27(6):1080–1094.
- [Jalali and Nadimi 2006] Jalali, A. A., and Nadimi, V. 2006. A survey on robust model predictive control from 1999–2006. In *2006 International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*, 207–207.
- [Keviczky et al. 2008] Keviczky, T.; Borrelli, F.; Fregene, K.; Godbole, D.; and Balas, G. J. 2008. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on Control Systems Technology* 16(1):19–33.
- [Li, Wendt, and Wozny 2002] Li, P.; Wendt, M.; and Wozny, G. 2002. A probabilistically constrained model predictive controller. *Automatica* 38(7):1171 – 1176.
- [Ono and Williams 2008] Ono, M., and Williams, B. C. 2008. Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In *2008 47th IEEE Conference on Decision and Control*, 3427–3432.
- [Pepy and Lambert 2006] Pepy, R., and Lambert, A. 2006. Safe path planning in an uncertain-configuration space using rrt. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5376–5381.
- [Richards and How 2004] Richards, A., and How, J. 2004. A decentralized algorithm for robust constrained model predictive control. In *Proceedings of the 2004 American Control Conference*, volume 5, 4261–4266 vol.5.
- [Schouwenaars et al. 2001] Schouwenaars, T.; Moor, B. D.; Feron, E.; and How, J. 2001. Mixed integer programming for multi-vehicle path planning. In *2001 European Control Conference (ECC)*, 2603–2608.