Fast Multi-Vehicle Path-Planning with Decentralized Heuristics

Aaron Huang

Prof. Brian Williams

Benjamin Ayton

**Abstract**

**Introduction**

The rapid integration of multiple autonomous vehicles into applications spanning industry and academia heralds an imminent need for architectures that support safe multi-vehicle coordination in a practical and optimal manner. These multi-vehicle coordination architectures must direct autonomous fleets toward the achievement of predefined goals while enforcing constraints. A critical element of multi-vehicle coordination is multi-vehicle path-planning in an environment where inter-vehicle interactions are possible. To be viable in practice, plans for autonomous fleets must be safe, optimal, and should have a reasonable time to computation. Unfortunately, optimization methods for multi-vehicle path-planning problems are often computationally intensive and can incur solution times that are exponential with respect to the number of vehicles and constraints.

This project aims to develop a heuristic method that reduces the time complexity of applying the Iterative Risk Allocation (IRA) optimization method to centralized multi-vehicle path-planning problems. The heuristic first approximates the naïve solution by using a Decentralized Model Predictive Control (DMPC) approach to separate the centralized problem into decentralized subproblems each considering a single vehicle. A fast IRA variant is used as the DMPC optimization method to approximate individual vehicle paths in the naïve centralized solution. We use the solution approximations to separate the centralized problem into smaller subproblems by identifying which sets of vehicles have significant interactions and should have coupled path-planning problems. Finally, applying IRA to each subproblem determined from the heuristic generates solution paths for all vehicles. We show that applying our heuristic method to the multi-vehicle path-planning problem significantly improves solution time compared to the centralized solution.

Our heuristic is primarily developed for coordination of a fleet of autonomous underwater vehicles (AUVs) for aquatic exploration. A typical expedition might consist of a multi-day venture where plans for multiple missions must be consecutively generated as quickly as possible to achieve maximum science gain. Mission objectives must be accomplished while ensuring temporal, collision, and other constraints are enforced. Running path-planning algorithms over the centralized planning problem incurs prohibitively long computation times that drastically hinder fleet utility and reduce science gained from the expedition. In this situation, our heuristic can decouple the centralized problem to enable more rapid solution computation time and achieve higher fleet utility over the duration of a research expedition.

(talk about how the heuristic can be extended to other types of vehicles with linear dynamics?)

**Literature Review**

Linear programming is an optimization method that represents mathematical models as linear programs (LP) where linear equations and inequalities of continuous variables describe the objective function, constraints, and properties. **(How, Schouwenaars, and De Moor)** demonstrate how mixed integer/linear programs (MILPs) can be used for multi-vehicle path-planning. MILPs are a variant on LPs that can represent obstacle and inter-vehicle collision avoidance constraints with the introduction of integer variables that allow constraints to be turned on or off. Highly optimized commercial software for nonconvex optimization can be used to solve path-planning problems formulated as MILPs to retrieve globally optimal vehicle trajectories given linear vehicle dynamics. Unfortunately, MILP optimization for large vehicle fleets have solution times that scale poorly because time complexity can be exponential as the size of the problem increases. This follows because the number of variables and constraints increase polynomially with the number of timesteps and number of vehicles **(Bertsimas and Tsitsiklis)**, rendering MILP-based path-planning intractable for longer missions involving many coupled vehicles. However, MILPs are still commonly used to model path-planning problems and form the mathematical basis for our method.

Model Predictive Control (MPC) is an online feedback control scheme that uses the receding horizon (RH) principle to calculate optimal trajectories over a limited number of timesteps into the future (referred to as the planning horizon). MPC performs an online trajectory optimization to retrieve an optimal control sequence at each time step. After executing the first control input of the control sequence, the same trajectory optimization and execution loop is recursively run on the updated vehicle state for all subsequent timesteps. In practice, a variant MPC scheme called Robust Model Predictive Control (RMPC) is more commonly used for path-planning. RMPC improves upon MPC by adding a disturbance at each timestep to model environmental uncertainties. RMPC techniques with bounded disturbances have been shown to generate trajectories through environments that are safe from obstacle collision at up to 3-sigma confidence at each timestep **(Pepy and Lambert 2006).**

Decentralized Model Predictive Control (DMPC) algorithms employ the strategy of decomposing the full multi-vehicle trajectory optimization problem into decentralized subproblems for each vehicle that are individually solved with an RMPC strategy **(Richards and How 2004)**. To account for inter-vehicle coupling constraints, each subproblem must be solved while considering all other vehicle trajectories. While DMPC offers significant solution time improvements over centralized methods, having fully interconnected vehicle couplings in each subproblem can still incur superfluous computation time for vehicle pairs that have no significant interactions. For example, there is no need for two vehicles moving in entirely opposite directions to have coupled planning problems. Additionally, many MPC methods are only feasible for path-planning problems in convex regions i.e. regions where no obstacles are present.

Methods to identify situations where vehicle coupling is necessary generally use informed heuristics to decouple unnecessary vehicle interaction constraints for decentralized control problems. **(Keviczky et al. 2008)** demonstrates a distance-based heuristic to maintain a communication topology graph that is updated over time, with undirected edges representing vehicle couplings. Undirected edges between any two vehicles indicate that both must account for the other’s actions when planning paths. A distance-based heuristic is used to determine vehicle couplings by evaluating the distance separating a pair of vehicles against a safe distance threshold. However, the practice of using heuristics to decouple subproblems for faster computation incurs a small but non-negligible risk of collision between decoupled vehicle pairs. **(Keviczky et al. 2008)** handles this by invoking emergency maneuvers computed via invariant set theory to prevent vehicle collisions. Our approach reduces the probability of infeasible trajectories by determining vehicle couplings using a more intelligent approach to identify probable vehicle interactions.

**Problem Statement**

The heuristic method we propose supports a path-planning system for autonomous underwater vehicles (AUVs) that uses the Iterative Risk Allocation (IRA) method to solve the path-planning problem described in **(Ono and Williams)**. Therefore, we will formally define the same problem and discuss components of the IRA method that are relevant to our heuristic approach.

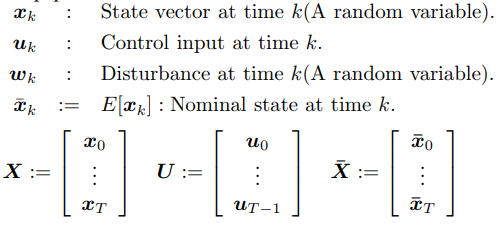
*RMPC Problem with Chance Constraints*

The problem described in the previous section can be represented as an RMPC problem formulated as a MILP. Traditional RMPC methods assume bounded disturbances on trajectory optimization problems. With bounds placed on the magnitude of disturbances, robust trajectories can be designed that are resistant to constraint failure against worst cases disturbances. However, 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 approaches.

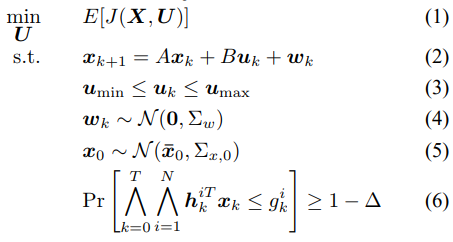
A different approach is to place *chance constraints* to limit the *probability* of violating constraints. There are two kinds of chance constraints – *individual* chance constraints limiting the probability of failure of a single constraint at a single timestep, and *joint* chance constraints limiting the probability of failure of any constraint in a problem **(Li, Wendt, Wozny)**. We formulate multi-vehicle path-planning as an RMPC problem with a joint chance constraint and unbounded environmental disturbances.

***Problem Definition***

*Notation*



*RMPC with a Joint Chance Constraint*

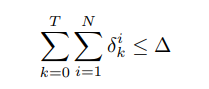


We wish to solve for a sequence of control inputs ***U*** that generate a state sequence ***X*** that minimizes the objective function in (1) while obeying the joint chance constraint shown in Equation (6). The joint chance constraint places a lower bound (1 – DELTA) on the probability that all constraints are fulfilled over all timesteps. In other words, DELTA directly represents the *risk* of any constraint failing over all timesteps. In Equation (2), state evolution from time *k* to *k + 1* is determined by applying control inputs *u\_k* and adding disturbance *w\_k.* The unbounded disturbance *w\_k* shown in Equation (4) is represented as a symmetric 2D Gaussian distribution. Adding *w\_k* in Equation (2) forces *x\_k+1* to also be unbounded. Therefore, guaranteed satisfaction for all constraints is impossible as the distribution of possible states for x\_k cannot be bounded. Equation (3) constrains control inputs and Equation (5) describes how *x\_k* is represented as a mean and covariance of vehicle state at time *k.*

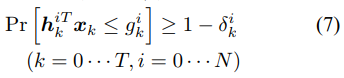
*Review of Iterative Risk Allocation*

Iterative Risk Allocation (IRA) is a two-stage optimization method developed by **(Ono and Williams 2008)** for RMPC problems with unbounded disturbances. An important concept supporting IRA is the usage of Boole’s inequality and an additional constraint to transform the joint chance constraint into a set of individual chance constraints (Blackmore, Ono, and Williams). More formally:

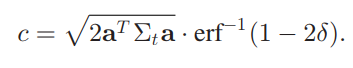
,



where delta\_ki represents the risk that the individual chance constraint i is failed at time k. These equations allow Equation (6) to be reformulated as the following:



Equation **(INSERT DELTA DECOMPOSITION EQUATION)** thus enables the concept of *risk allocation,* wherein a portion of the total risk DELTA can be allocated to risks delta\_ki for individual chance constraints. In the context of obstacle avoidance constraints, (Blackmore, Ono, and Williams) provide a critical contribution for convex obstacle avoidance constraints with the following deterministic equation:



This offers an intuitive relationship between delta and the minimum buffer distance *c* a vehicle must be from an obstacle to obey the corresponding individual chance constraint. As delta increases, the buffered distance *c* decreases as more risk is allocated towards the individual chance constraint.

IRA uses an iterative two-stage optimization procedure to find the optimal risk allocation over all individual chance constraints. The general idea of the risk reallocation upper stage is to move risk from *inactive* constraints (that have probability of failure far below their allocated risk) to *active* constraints (probability of failure close to their allocated risk) to monotonically decrease overall cost. This new risk allocation defines new obstacle buffer zones that the lower stage solves the RMPC problem with (typically using a commercial MILP solver). New probabilities of failure for all constraints are computed, starting another iteration of risk reallocation.

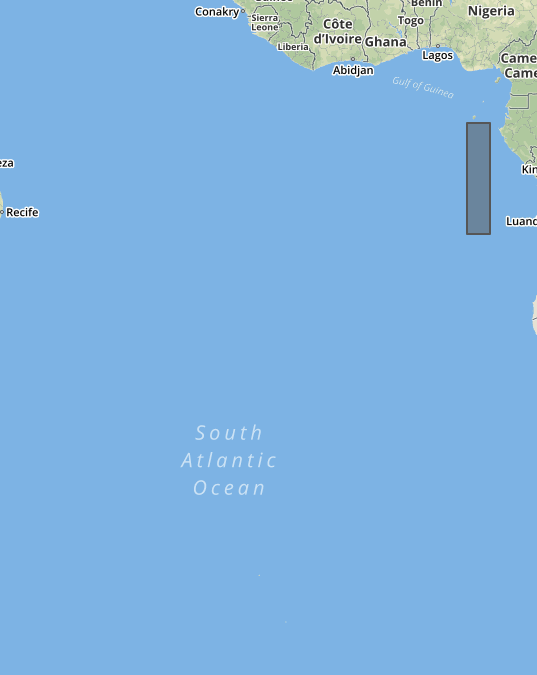
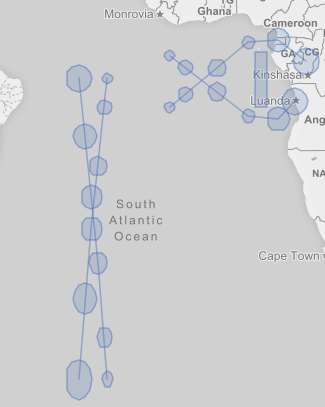
Although the specifics of the iterative procedure is beyond the scope of this paper, it is important to note that the optimal trajectories determined from the lower stage can be used to determine a probability of collision between a vehicle and any obstacle at any timestep. This plays an important role in the determination of vehicle interactions described shortly.

**Technical Approach**

When formulated as a multi-vehicle RMPC problem with unbounded disturbance, multi-vehicle path-planning has a solution time that grow exponentially with the number of vehicles being planned over. However, many problems do not require vehicle coupling constraints for every vehicle pair. In this section, we will introduce and examine such a problem and argue that solution times can be generally improved by separating the set of vehicles into smaller subsets of vehicles with *relevant* coupling constraints.

**Multi-Vehicle Path-Planning Problem**

Consider a path-planning problem with **(INSERT NUMBER OF VEHICLES = N)** vehicles in a nonconvex environment as shown in **(INSERT FIGURE NUMBER)**. The naïve approach to solving this problem would be to apply IRA to a centralized RMPC problem with coupling constraints on every vehicle pair. The output of such an approach is shown in **(INSERT FIGURE NUMBER)**

**To be replaced with better examples – figure on left will have vehicle start nodes in green, goal nodes in red, with letters distinguishing starts and goals for a vehicle. Figure on right will show naïve solution**

Visual inspection of the solution suggests that is not necessary to have vehicle coupling constraints for all vehicle pairs. The two vehicles that cross on the left do not need to account for the two trajectories of the two vehicles that are planned on the right. Since problem complexity increases exponentially with the number of vehicles, it would be far more efficient to decompose the set of four vehicles into two subsets each containing two vehicles to be individually solved over for less total runtime.

**Heuristic Method**

We propose a heuristic method that determines subsets of vehicles with relevant coupling constraints in complex multi-vehicle RMPC problems and demonstrate how they can be tractably computed using our approach. The proposed heuristic method is a preprocessing step that fuses a DMPC approach with risk allocation concepts to approximate paths generated by the naïve, centralized approach. Coupled vehicle subsets can then be determined by examining the solution approximation for significant vehicle interactions.

*Decentralized Model Predictive Control (DMPC)*

The core of the heuristic is a solution approximation that must be similar to the naïve solution while incurring far less runtime. As solving multi-vehicle RMPC problems is computationally exponential with the number of vehicles, we divide the centralized problem into N chance-constrained subproblems, each optimizing the trajectory for a single vehicle. At every timestep, the subproblems are solved in a predetermined vehicle ordering sequence using a fast variant of IRA. Fast IRA consists of one iteration where the lower stage optimizes a trajectory given an initial risk allocation that is uniform among all constraints. After first control input is applied, the heuristic method moves onto the next vehicle’s subproblem.

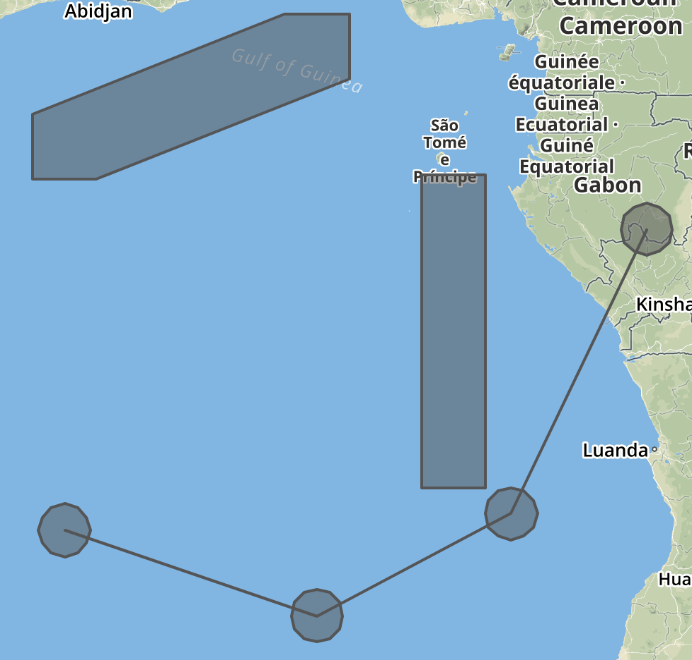
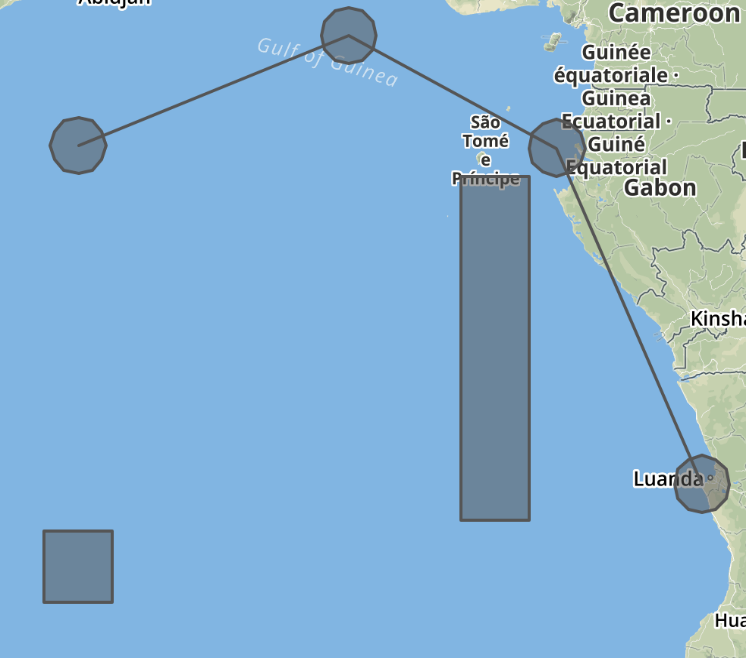
A key difference from **(Richards and How 2004)** is that our approach removes all vehicle coupling constraints from each decentralized subproblem and does not explicitly account for the actions of other vehicles. Instead, we introduce a novel method that implicitly represents vehicle actions by creating obstacles that geometrically bound likely vehicle states at that timestep (hereafter referred to as *vehicle obstacles*).

*Vehicle Obstacles*

A vehicle obstacle formally bounds the 2-sigma confidence region of a vehicle’s state. Assuming symmetric 2D Gaussian disturbances act on vehicle states, the radial distance *d* bounding the 2-sigma confidence region of a state distribution x\_t is calculated as follows:

**INSERT MATH SHOWING CALCULATION OF DISTANCE d FOR 95% CONFIDENCE REGION**

As truly circular objects are difficult to represent, *d* is instead used to compute square coordinates around *x\_t*. Vehicles in goal and vehicles at *t = 0* that haven’t yet been planned have square obstacle representations bounding the 2-sigma confidence area at the current timestep. Otherwise, the vehicle obstacle is generated by wrapping a convex hull around the square coordinates generated at *x\_t* and *x\_t+1*. Figures (**INSERT FIGURE NUMBERS**) respectively show examples of a square vehicle obstacle at *t = 0* and a larger vehicle obstacle bounding the states from *t = 0* to *t = 1.*



**\*note to graders\***: I am looking into ways to plot these geoJSON files without overlaying them on the Earth

Vehicle obstacles enable the implicit communication of vehicle intent without incurring the increase in computation time due to coupling constraints. A key benefit of representing vehicle as obstacles is that we can easily compute the probability of collision from a vehicle to all vehicle obstacles as an intuitive metric for vehicle interactions.

*Tracking Risk*

We track risk by maintaining a risk pool DELTA\_k representing the remaining risk for the mission at time k. At the beginning of each timestep, DELTA\_k is distributed equally to the N joint chance constraints of the remaining subproblems. After sequentially solving every subproblem using a typical convex solver, the sum of risks allocated to the individual chance constraints at time k of every subproblem is subtracted from DELTA\_k to find DELTA\_k+1.

**INSERT MATH SHOWING RISK POOL CALCULATION**

**Separating the Centralized Problem with Vehicle Interactions**

To separate the centralized multi-vehicle RMPC problem, we determine the subsets of vehicles that have significant vehicle interactions and must be planned with coupling constraints. Fortunately, we can easily retrieve a metric for vehicle interactions by recording the probabilities of vehicle obstacle collisions for each vehicle pair over all timesteps.

Initially, each vehicle belongs to a subset that contains itself. After the solution approximation is complete, we evaluate each vehicle pair by comparing the probabilities of collision against a threshold value past which we deem the vehicle pair to have significant vehicle interactions. If a vehicle pair has significant interactions, then we merge the two coupled sets either vehicle is in to generate a larger coupled set.

After generating the coupled sets, we decompose the original centralized multi-vehicle path-planning problems into a set of smaller subproblems that each contain vehicles from one of the coupled sets. Finally, we run the primary path-planning solution on all subproblems and retrieve our solution to the centralized problem by extracting the solved vehicle trajectories.

**The Algorithm**

(INSERT PSEUDOCODE FOR MULTI-VEHICLE)

(INSERT PSEUDOCODE FOR SEPARATE-PROBLEM)

**Results and Analysis**

In this section, we demonstrate the results of applying our heuristic-based approach to multi-vehicle path-planning problems in simulation. We examine paths generated by the naïve solution along with paths generated by our solution approximation and discuss causes of discrepancies. We then investigate how the centralized problem is decoupled and compare the average runtime of solving the multi-vehicle path-planning problem with and without the heuristic.

**Solution Approximation**

In this example, we demonstrate the application of our approach on a multi-vehicle RMPC problem with **N** vehicles

(generate solution approximation graph and compare)

**Simulation Testing**

To compare the average runtime of our approach against that of the naïve approach, we randomly generate and solve problems in simulation while recording solution times. Each sample problem generates random start and goal poses for every vehicle being considered and uses a simple environment with a single obstacle for simplicity. The following parameters are used:

(**IN LATEX**): 12 timesteps, 0.1 risk, 0.0001 threshold, 2 to 8 vehicles

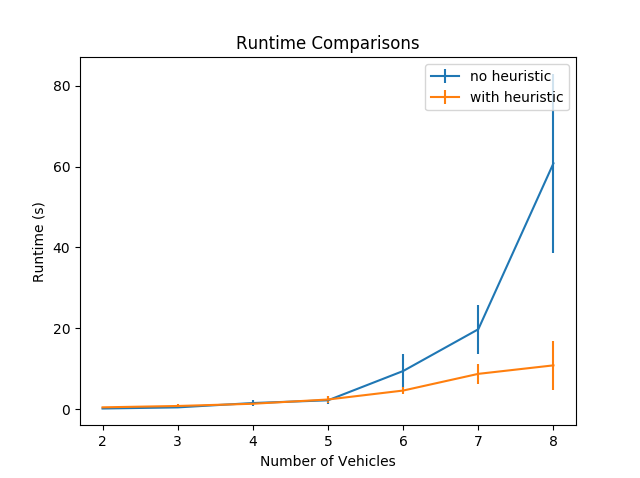


Figure **INSERT NUMBER** demonstrates a significant reduction in solution time when solving complex RMPC problems with the heuristic, *even* when DMPC solution approximation runtime is included. Simulating 100 sample problems with more than 8 vehicles proved to be too computationally intensive given our time constraints, but it is evident that our method will run faster by larger margins as the number of vehicles being considered increases. Much of this improvement can be attributed to the fact that most randomly generated complex problems can be separated into small, simple subproblems with one or two vehicles each. Figures **INSERT FIGURE NUMBERS** depict such a

Additionally, our approach exhibits standard deviation that is far smaller than the standard deviation generated from the naïve solution. This follows because of the naturally larger deviations in solution computation time for more complex problems. This was especially evident when we solved problems with more than 8 vehicles, with solution times ranging from 10 to 100 seconds.

Our

**Future Work**

Future improvements to our method would allow for more accurate solution approximation. A major improvement being added is the notion of dynamic vehicle obstacles. Our current vehicle obstacle representations are static - decentralized problems are solved under the assumption that vehicle obstacles do not change location over all future timesteps. Dynamic vehicle obstacles will result in more accurate solution approximations and reduce the number of failed solution approximations by ensuring that vehicle obstacles cannot permanently block goal locations if there happens to be an overlap.

Our current implementation solves the subproblems in the same vehicle ordering at every timestep. This constrains vehicles later in the ordering more tightly since preceding vehicles may have taken a greedier route that might not reflect its trajectory in the true concurrent situation. Randomizing the order in which decentralized problems are solved at each timestep could eliminate the inherent advantage that vehicles earlier in sequence accumulate over timesteps. Additionally, the initial risk allocation for each decentralized problem can be made more intelligently. For instance, running a single iteration of risk reallocation and using the new risk allocation to initialize the next timestep’s decentralized problem might induce more accurate solution approximations.

At a more general level, further investigation into using heuristics to decouple multi-vehicle planning-problems could consider using faster path-planning algorithms like the visibility graph method to approximate solutions even more quickly. However, consideration must be given to the tradeoff between solution approximation accuracy and runtime and heuristics should be tailored to the application at hand.

**Conclusion**

We have presented a novel heuristic-based method that intelligently decouples computationally expensive multi-agent RMPC problems to reduce solution time complexity. Our approach uses a Decentralized Model-Predictive Control approach with vehicle obstacle representations to determine sets of coupled vehicles that have *relevant* vehicle coupling constraints. Applying more complete path-planning methods to the subproblems for each coupled set generates collision-free paths in far less time than the naïve brute-force approach.

We envision this heuristic being used to enable more rapid planning capabilities for vehicle swarms being used in the field. Although this heuristic was specifically designed to support autonomous underwater vehicle exploration, fast multi-vehicle path-planning spans a large range of academic, industrial, and emergency applications. Further research into fast multi-vehicle path-planning will facilitate more rapid integration of effective multi-vehicle coordination capabilities in situations where they are sorely needed.

**References**

[1] Schouwenaars, T., et al. “Multi-Vehicle Path Planning for Non-Line of Sight Communication.”*2006 American Control Conference*, 2006, doi:10.1109/acc.2006.1657643.

[2] Ono, M., & Williams, B.C. (2008). Iterative Risk Allocation: A new approach to robust Model Predictive Control with a joint chance constraint. *CDC*.

[3] Ono (2012). Robust, Goal-directed Plan Execution with Bounded Risk

[4] Alexis, Kostas, et al. “Robust Model Predictive Flight Control of Unmanned Rotorcrafts.”*Journal of Intelligent & Robotic Systems*, vol. 81, no. 3-4, 2015, pp. 443–469., doi:10.1007/s10846-015-0238-7.

A. A. Jalali and V. Nadimi, “A survey on robust model predictive control from 1999–2006,” in Proc. Int. Conf. Comput. Intell. Modell., Control Autom., 2006.

Dimitris Bertsimas and John N. Tsitsiklis. Introduction to Linear Optimization. Athena Scientific, 1997.

R. Pepy and A. Lambert, “Safe path planning in an uncertain-configuration space using RRT,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2006, pp. 5376–5381.

Pu Li, Moritz Wendt, and Gunter Wozny. A probabilistically constrained model predictive controller. Automatica, 38:1171–1176, 2002.

L. Blackmore and M. Ono, “Convex chance constrained predictive control without sampling,” in Proc. AIAA Guid., Navigat. Control Conf., 2009.