Heuristics for Efficient Multi-Vehicle Coordination

Model Based Embedded and Robotic Systems Group (MERS)
Aaron Huang (ahuangg@mit.edu)
Brian C. Williams (williams@csail.mit.edu)
Benjamin J. Ayton (aytonb@mit.edu)

Abstract

Multi-vehicle coordination plays a crucial role in many commercial and public endeavors like transportation, search and rescue, and robotic exploration. As autonomous vehicles percolate through more industries, multi-vehicle coordination architectures that maximize utility while enforcing various constraints will need to be improved upon. Unfortunately, concurrently planning safe trajectories for every agent in a fleet of vehicles poses a computationally expensive problem with an exponential solution time with respect to the number of vehicles being coordinated. Current multi-vehicle coordination architectures employ distributed path-planning algorithms to reduce the complexity of the problem. However, many distributed algorithms can still retain prohibitively expensive subproblems that impede computation speed. This project will develop fast heuristics to quickly determine which vehicles to cut out of a path planning subproblem, ideally reducing solution complexity from exponential time to pseudo-polynomial time. We explore the "Visibility Graph" and "Single-step Receding Horizon Approximation" strategies and discuss the pros and cons of each. We hope to develop a multi-vehicle architecture for the MERS Scouts project that would ideally see implementation on a 2019 science mission.

I. Problem Statement & Motivation

Autonomous vehicles are preferable in many situations like search and rescue or space exploration where establishing communication and direct control of a fleet of vehicles is time-consuming, costly, and prone to human error. As fleets of multiple autonomous vehicles are increasingly integrated into applications spanning many industries, architectures that support safe coordination of multiple vehicles in a practical and efficient manner need to be developed. These multi-vehicle coordination architectures must maximize utility while concurrently obeying directed constraints. A critical element of safe multi-vehicle architectures is path-planning.

Initial approaches to multi-vehicle path-planning encoded the entire multi-vehicle path-planning problem into a single mixed integer/linear program (MILP). MILPs are comprised of a cost function to be minimized and linear equations and inequalities encoding constraints on vehicle dynamics, obstacle avoidance, and other necessary factors. MILPs are "mixed" linear programs because the domains of variables used in MILPs are a mix of continuous or integer numbers. Unfortunately, time complexity for solving a centralized MILP is computationally exponential with respect to the number of vehicles being considered and becomes prohibitively expensive for most practical applications. The full path-planning solution that MERS currently uses for robotic underwater exploration (referred to as pSULU) is a single-vehicle solution that runs the Iterative Risk Allocation (IRA) algorithm on a MILP representing the path-planning problem (further explained in "Related Works"). If we were to naïvely plan for the entire fleet by encoding every single vehicle into a single MILP to solve, computation time would be very exponential, rendering the problem intractable.

More recent solutions for multi-vehicle path planning have used distributed planning algorithms to divide the full problem into more localized subproblems that each have drastically reduced solution time complexity. Optimal distributed planning algorithms need to solve each smaller subproblem and combine the results to retrieve a result that is very similar to the correct result. Ideally, a smaller path-planning subproblem would prune the fleet and only incorporate vehicles that travel close enough to each other to have coupled paths. However, the heuristics used to determine appropriate vehicle-subproblem couplings are often very shallow or are not even described in literature, indicating the need for development of a better heuristic.

For this SuperUROP project, I will develop fast and effective heuristics that use more comprehensive vehicle information to determine appropriate vehicle-subproblem couplings, with the goal of reducing solution time complexity from exponential to pseudo-polynomial time. The heuristic will do this by quickly approximating the final solution generated by the full path-planner and evaluating the resulting paths to identify which vehicles need to be coupled. In this proposal, we will describe the "Visibility Graph" and "Single-step Receding Horizon Approximation" solution architectures being considered and discuss the pros and cons associated with each.

Successful development of a useful heuristic would expedite the development of an efficient multi-vehicle coordination architecture for the MERS robotic undersea exploration program (referred to as Scouts). Ideally, a useful heuristic would substantially reduce computation time required for safe trajectories to be found in a multi-vehicle architecture, enhancing science gathering capabilities on a NASA funded Scouts field campaign to map out the Columbo volcano scheduled for 2019.

II. Related Work

Since the heuristic that will be developed for this project needs to be a fast approximation of the current path-planner, we will discuss recent work in path-planning algorithms. One of the first solutions to multi-vehicle path-planning pioneered the usage of MILPs to find fuel-optimal paths for multiple vehicles [1]. Mixed integer/linear programs were demonstrated to be useful because integer variables give us the capability to reasonably encode obstacle avoidance constraints.

At a high level, vehicle path-planning problems encoded in MILPs can be solved with a Robust Model Predictive Control (RMPC) strategy [5]. RMPC is commonly used to control dynamic systems through an iterative, finite-horizon optimization of the MILP cost function that still obeys given constraints. The general solution strategy involves planning control inputs over a finite number of timesteps into the future, executing a single timestep's control input and repeating. Thus, solutions are iteratively found using a "receding horizon" approach that repeatedly plans and executes until the goal is found.

The fact remains that centralized MILPs quickly become intractable as we scale the number of vehicles. Alternative solution methods revolve around a strategy of breaking the full path planning problem into smaller, more computationally efficient subproblems each coordinating subsets of vehicles that interact only with each other, decoupling the subset from the entire fleet [2]. For our purposes, it is important to note that many distributed planning architectures do not describe the specific process by which vehicles are separated into subproblems. We wish to develop a heuristic that approximates the full path-planning algorithm we use for Scouts.

Prior arts have included the ellipsoidal relaxation approach [3] and Particle Control [4], but Scouts relies upon the two-stage optimization method for RMPC called Iterative Risk Allocation (IRA) [5] that forms the basis of pSULU, our full path-planning solution. At a high level, IRA introduces the notion of "total risk" given for a mission and iteratively solves for the best allocation of risk over all constraints that minimizes our cost function. The algorithm assigns a portion of the total risk to each constraint and calculates the probability of constraint failure given the assigned risk. For all constraints with a probability of constraint failure lower than the allocated risk, each is identified to be either active or inactive depending on how close the allocated risk is to the minimum calculated acceptable risk. Risk is then reallocated from inactive constraints to active constraints, at which point we iterate until a solution is found. A visualization of IRA is depicted below on the right – notice how the algorithm decides to take more risk around the sharp corner in return for less actuation cost and a quicker path to the goal.

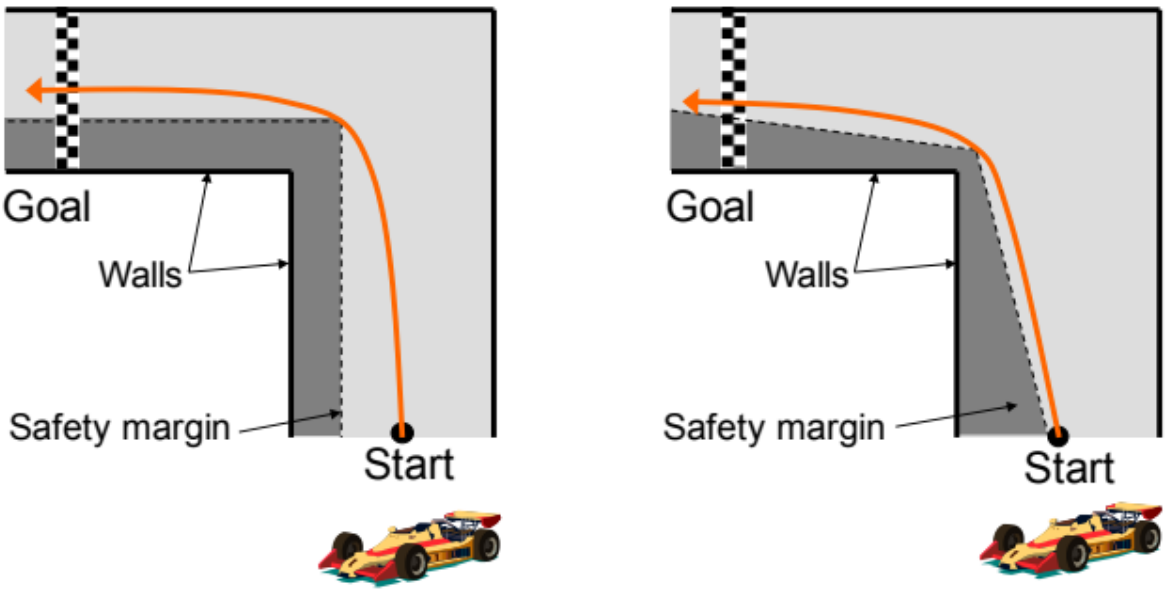

Figure 3: Visualization of IRA versus uniform risk allocation (Williams, 16.412 notes)

III. Technical Approach

Contemporary multi-vehicle path-planning architectures demonstrate the ubiquity of distributed planning algorithms. The strength of distributed planning techniques lies in the ability to reduce overall computation time by breaking the full path-planning problem into smaller, decoupled subproblems. We wish to develop a heuristic to separate the full problem into more tractable subproblems with the end goal of building a multi-vehicle path-planning architecture that uses pSULU as the primary path-planner. This heuristic acts as a "first-pass" filter that roughly approximates the pSULU path planning solution without executing a full run of the IRA algorithm. Path-approximations generated by the heuristic are then used to separate vehicles into appropriate subproblem couplings that are solved and combined to form a full solution. To this end, we propose a tentative solution strategy that revolves around using quick path approximations to decide which "safe pairs" of vehicles can be planned for independently. There are two solutions strategies we will explore.

Visibility Graph Approximations

We can use visibility graphs to quickly approximate the path of a vehicle to its goal. Visibility graphs are very computationally inexpensive to compute and are very simple to implement. From a start node, lines are extended to every visible vertex on all polygons surrounding the start. We iterate and draw lines from vertex to vertex until we reach the goal. Paths are then found using the A* search method. We can run through all pairs of vehicles to determine which vehicles need to be coupled depending on how close their paths are.

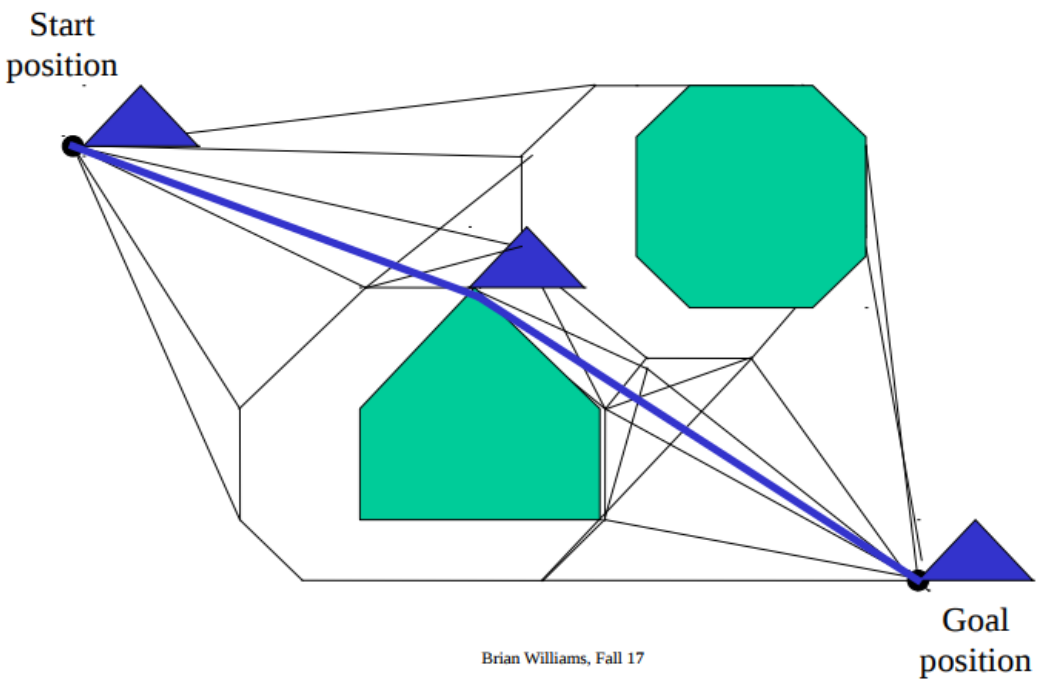

Figure 4: Visualization of a visibility graph (Williams, 16.413 notes)

The figure depicted above is a basic representation of how a visibility graph works. The green polygons represent environment obstacles that the blue triangle representing the vehicle must avoid. The black lines are candidate paths drawn by the visibility graph with the blue line representing the solution path.

Unfortunately, the visibility graph method is not without its flaws. For one, visibility graphs do not have any notion of vehicle size, dynamics, or allocated risk. Consequently, paths planned through a visibility graph method might be entirely different from paths planned by IRA and might be impossible for the vehicle to traverse. This is particularly concerning when considering the possible case where a visibility graph-based heuristic might dub two vehicles as a safe pair while paths planned using IRA lead to an intersection. Ideally, we would like to place conservative guarantees on our heuristic where if the heuristic passes, the vehicles will not intersect. Should the heuristic fail, we would still expect the possibility to remain that the vehicles do not intersect.

Single-step Receding Horizon Approximations

An alternate solution we recently conceptualized involves running a single iteration of pSULU at each timestep in a receding horizon fashion. At every iteration, each vehicle treats the other vehicles as static obstacles added to the environment. Then, a single iteration of pSULU is run for each vehicle. Each vehicle acts upon only the calculated control input for the current time step. After every vehicle has moved for a single control input, we can redraw the environment and update every vehicle's position as a new static obstacle. We continue to iterate until each vehicle reaches its goal.

In this way, we proceed in an iterative receding horizon fashion that uses a single iteration of the same path-planner we are trying to approximate at every update. The clear benefit of this heuristic is that we can expect the paths approximated by this heuristic to be more analogous to the complete pSULU solution. However, the pressing issue remains of how long such a heuristic would take to run and how that weighs in comparison to the quality of the solution. Since we have also never attempted a solution like this before, there is always the probability that unforeseen issues will arise that complicate implementation, draining our time and effort.

Metrics

Evaluating the tradeoff between heuristic runtime and optimality of the solution can be considered an advanced metric that we would like to optimize sometime in the future. However, at our current stage we are focused on simply determining which vehicles to plan centrally or separately based on our heuristic. In this case, our major evaluation metric is run time/complexity.

We can calculate run time by measuring the combined time of running planning with the heuristic versus planning centrally without the heuristic. We would ideally show that total run time is shorter when the heuristic finds separable problems in randomly generated missions. Complexity would be derived from a theoretical examination of the heuristic algorithm. If we can demonstrate that the heuristic has a pseudo-polynomial run time, it will minimally affect the complexity of the full path-planning system when applied. Lastly, we can evaluate the heuristic's failure rate by determining how often it fails to detect a separable problem and how often it incorrectly deems a problem to be separable.

IV. Conclusion

Although both solution strategies are viable, we will likely move forward with implementing the heuristic using the "Single-step Receding Horizon Path Approximation" architecture for its similarity to the full path-planner and interesting solution strategy. Should the heuristic prove useful, we will then move towards using the heuristic to develop a full multi-vehicle coordination architecture for Scouts.

Multi-vehicle coordination architectures will play an increasingly important role in our world as autonomous vehicles become more capable and the free market embraces the technology. Multi-vehicle coordination is also a fantastic tool for research and development in many fields for its ability to greatly enhance and deepen sensing capabilities in remote and inhospitable environments. Hopefully, a successful project would reduce path computation time for autonomous fleets, eroding one weakness of multi-vehicle architectures and promoting their spread to more areas of research, development, and industry.

V. Tentative Timeline

October	Formally submit detailed proposal, conceptualize and study background required for effective heuristic implementation
November	Expand upon visibility graph heuristic to account for vehicle dynamics, actuation, and total allocated risk, develop oral and visual presentation
December	Present proposal to sponsors
January	MERS goes on research trip to Hawaii, I will stay in Cambridge for an industry internship, stay in touch with lab members
February	TBD

VI. References

[1] Moor, B.D., & Schouwenaars, T. (2001). Mixed Integer Programming for Multi-vehicle Path Planning.

[2] Liu, Jinfeng & Chen, Xianzhong & Munoz de la Pena, David & Christofides, Panagiotis. (2010). *Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. Part I: Theory*. Proceedings of the 2010 American Control Conference, ACC 2010. 3148 - 3155. 10.1109/ACC.2010.5531017.

[3] Dennis Harald van Hessem. *Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation.* PhD thesis, Delft University of Technology, 2004

[4] Lars Blackmore. A probabilistic particle control approach to optimal, robust predictive control. In *Proceedings of the AIAA Guidance, Navigation and Control Conference,* 2006.

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