Task Assignment and Trajectory Planning in Dynamic environments for Multiple Vehicles

Abstract— **We consider the problem of finding collision-free trajectories for a fleet of automated guided vehicles (AGVs) working in ship ports and freight terminals. Our solution computes collision-free trajectories for a fleet of AGVs to pick up one or more containers and transport it to a given goal without colliding with other AGVs and obstacles. We propose an integrated framework for solving the goal assignment and trajectory planning problem minimizing the maximum cost over all vehicle trajectories using the classical Hungarian algorithm. To deal with the dynamics in the environment, we refine our final trajectories with CHOMP (Covariant Hamiltonian optimization for motion planning) in order to trade off between path smoothness and dynamic obstacle avoidance.**

Keywords: *Multi-robot, task assignment, path planner*

I. Introduction

There are many ways to address the problem of motion planning for multiple robots. There are generally two approaches for solving this issue - centralized and decentralized. In centralized approach, the configuration spaces of the individual robots are combined into one composite configuration space which is then searched for a path for the whole composite system. In contrast, the decoupled/decentralized approach computes separate paths for the individual robots and then resolves possible conflicts of the generated paths. Though the former approach produces a complete and optimal solution, the time complexity is very huge. However, in case of the other, the solution is far from optimality. For our intended application, our approach falls in between the two categories which is a decentralized system that is complete and guarantees optimality. The inspiration of this approach comes from the recent researches in multi robot coordination and task allocation methods which have adapted methods from operational research community. Here, the methods used for task assignment for a group of workers to goals has been adapted to multi-robot task allocation and trajectory planning problems as in [1]. By finding an optimal path and modifying it to be collision-free reduces the complexity for multi vehicle planning. Hence, coupling goal assignment with trajectory planning is more advantageous compared to the traditional two approaches in multi-vehicle path planning. The computational complexity of planning collision-free trajectories for multiple robots typically grows exponentially with the number of robots. This approach has recently been adapted by many researchers as in [1], [8].

II. SPADES Framework

Fig. 1 shows our planning framework, which we refer to as SPADES in this paper, short for *Simultaneous Planning and Assignment in Dynamic EnvironmentS*. Our framework consists of three main parts: task assignment, global path planning and trajectory planner. Our approach takes as input a set of goals that should be allocated to a set of available vehicles – e.g. in a container terminal, these goals are the locations where containers should be picked up and taken by the available vehicles. Then, an optical assignment planner assigns vehicles to the nearest goals such that a overall total path cost is minimum. A global planner plans obstacle-free paths for each of the vehicles. The resulting trajectories are later continuously refined by a trajectory planner to avoid collisions with dynamic obstacles or static obstacles as well as to check for conflicts with other paths, treating all other vehicles as moving obstacles. Algorithm 1 shows our integrated planning framework with the three steps.

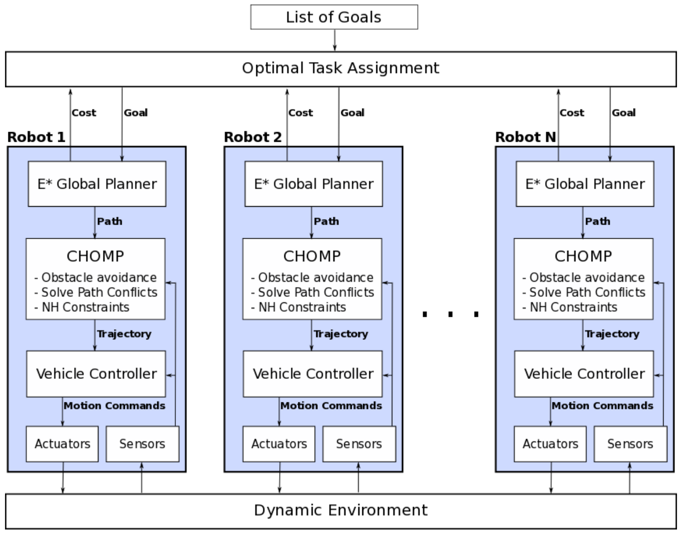

Fig. 1. SPADES Planning Framework.

In the rest of this section we describe each step of our planning framework in detail.

Algorithm 1: SPADES Planning Framework.

Data: global map, local maps, localization data
Result: list of obstacle-free trajectory points
Run Simultaneous Planning and Assignment
while $T = 1 : N$ **do**
  for *each vehicle* v_i **do**
    Run CHOMP (v_i path, $localMap_i$)
    Execute trajectories from $T - 1$ to T
  end
end

A. Assignment

The Hungarian Algorithm is adopted as our task assignment method. This approach solves the linear assignment problem with computational complexity bounded polynomially for N robots, $O(N^3)$, and is the most efficient known method to optimally solve the linear task assignment problem. E* provides the costs of assignment given as the path length from the shortest path between goals and starting locations computed with the E* path planner algorithm. To this end, we compute a cost matrix of the entire site map which contains the path lengths in the matrices between the containers and the vehicles. This cost matrix is generated by using a distance transform of the E* algorithm which calculates the Euclidean distance of the path length between them. This cost matrix is then fed in to the assignment module which calculates the optimal assignment using the Hungarian algorithm as it is detailed in Algorithm 2.

Algorithm 2: Simultaneous Planning and Assignment

Data: Input site map with localization data
Result: Optimal task assignment
Calculate list of to-schedule containers c_j and its goals g_k **Calculate** list of available vehicles v_i
for *each* c_j *in containers* **do**
  for *all vehicles in availistvehicles* av_i **do**
    Compute E* path from c_j to all av_i
    return dist[A_j]=dist[c_j] to all *vehicles*
  end
end
for *each* c_j *in containers* **do**
  Compute E star distance transform from c_j to g_k
  return dist(B_j)=dist(c_j) to its g_k
end
for *each* c_j *in containers* **do**
  for *each vehicle in* av_i **do**
    total path cost of c_j = dist(A_j) + dist (B_j)
    return *cost matrix*
  end
end
run task assignment algorithm
return *optimal task assignment*

B. Path Planner

To assign containers to available vehicles we employ an optimal task assignment approach together with E*[2]. The E-star algorithm is a path planner which supports dynamic replanning and path cost interpolation, resulting in lightweight repairing of plans and smooth paths during execution. The algorithm computes C-space samples of a crossing-time map defined by the monotonic expansion of a continuous closed surface or a contour from the goal through the environment By modulating the propagation speed in function of environment characteristics, i.e. the effort of traversing certain regions, the crossing-time map becomes a navigation function that reflects the influence of the environment on the optimal-path of the robot, and it suffices to follow its steepest gradient from any point to drive the robot to the goal. Thus, E* is employed to provide assignment costs in the form of the path lengths computed with this approach to our optimal task assignment algorithm.

C. Trajectory Refinement

CHOMP is a method for trajectory optimization invariant to re-parametrization that iteratively improves the quality of an initial trajectory, optimizing a functional that trades off between a smoothness and an obstacle avoidance component.

The goal of CHOMP is to iteratively improve the quality of an initial trajectory by optimizing a functional that trades off between a smoothness and an obstacle avoidance component. Thus, this approach is useful in dealing with path confliction and dynamic obstacles.

III. Conclusions

In this paper we introduced SPADES, an integrated framework for solving the goal assignment and trajectory planning simultaneously, suited for dynamic environments. This integrated framework for combined task and motion planning helps in dealing the problem of multi vehicle planning effectively. Future work include performing real world experiments with two AGVs platforms. We also envision to extend our approach to plan in belief space [9] instead so as to add robustness against uncertainty in robot perception.

References

[1] Matthew Turpin, Karthik Mohta, Nathan Michael, Vijaya Kumar. Goal Assignment and Trajectory Planning for Large Teams of Aerial Robots *Autonomous Robots* Dec 2014.

[2] Roland Philippsen and Roland Siegwart. Smooth and efficient obstacle avoidance for a tour guide robot. *In Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2003.

[3] Sean Quinlan and Oussama Khatib. Elastic bands: connecting path planning and control.*In Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1993.

[4] Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. *In Proc. Int. Conf. Robotics and Automation (ICRA)*, 2009.

[5] Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. *Springer Tracts in Advanced Robotics, Robotics Research: The 14th International Symposium (ISRR)*, 70, 2011.

[6] Lantao Liu, Dylan A. Shell, Nathan Michael. From Selfish Auctioning to Incentivized Marketing *Autonomous Robots* Dec 2014.

[7] Lantao Liu, Dylan A. Shell, An anytime assignment algorithm: From local task swapping to global optimality.*Autonomous Robots* 35: 271-286, 2013.

[8] Jingjin Yu and M LaValle, Distance optimal formation control on graphs with a tight convergence time guarantee. *In Decision and Control (CDC)*, 2012 IEEE 51st Annual Conference on, pages 4023-4028, 2012.

[9] R. Valencia, M. Morta, J. Andrade-Cetto, and J. M. Porta, Planning reliable paths with Pose SLAM. *IEEE Transactions on Robotics*, 29:4, pages 1050-1059, 2013.