Robert Effinger

Final Project Report

Cognitive Robotics

16.412J - Spring 2004

May 13, 2004

Table of Contents

1.) Table of Contents 2

2.) Overall Description 3

a.) Brief Group Project Description

b.) Individual Accomplishments

3.) Create Framework for TPN to be solved as a Conditional CSP 5

a.) Definition of a Conditional CSP

b.) Implementation Example - Two Functions

i.) getNextDecisionStartNodes( )

ii.) deactivateNestedDecisionStartNodes( )

c.) Pseudo-code and function explanations

4.) Dynamic Backtracking Implementation within Kirk 8

a.) Overall Description of Dynamic Backtracking

b.) Pseudo-code

c.) Insights from Testing Dynamic Backtracking on Representative TPNs 11

5.) Interfaces with Generative Planner and Roadmap Path Planner 12

a.) Roadmap Path Planner

i.) A Macro to update lower activity time-bounds

b.) Generative Planner

ii.) Passing TPN structure by pointer

6.) Branch and Bound w/ Dynamic Backtracking for Anytime and Optimal Search

7.) Summary and Conclusions 16

2.) Overall Description

2.a) Brief Group Project Description

For the Final Group Project in 16.412J, myself, Dan Leaute, Seung Chung, and Dan Lovell developed an autonomous cooperative UAV demonstration using the Cloud-Cap Autopilot Airplane Simulator. Our group devised a scenario and map in-which a team of UAVs can cooperatively plan to extinguish forest fires. The goal of this project is to create a cooperative multi-UAV activity planner that only needs as input high-level mission goals and specifications(such as put-out-fire or image-fire-damage). The planner should then be able to decompose the high-level mission goals into lower-level goals, and motion commands, and then execute them autonomously. Allowing the operator to specify high-level mission goals instead of sending detailed sequential commands to the UAVs reduces planning time and operator error, and increases plan flexibility.

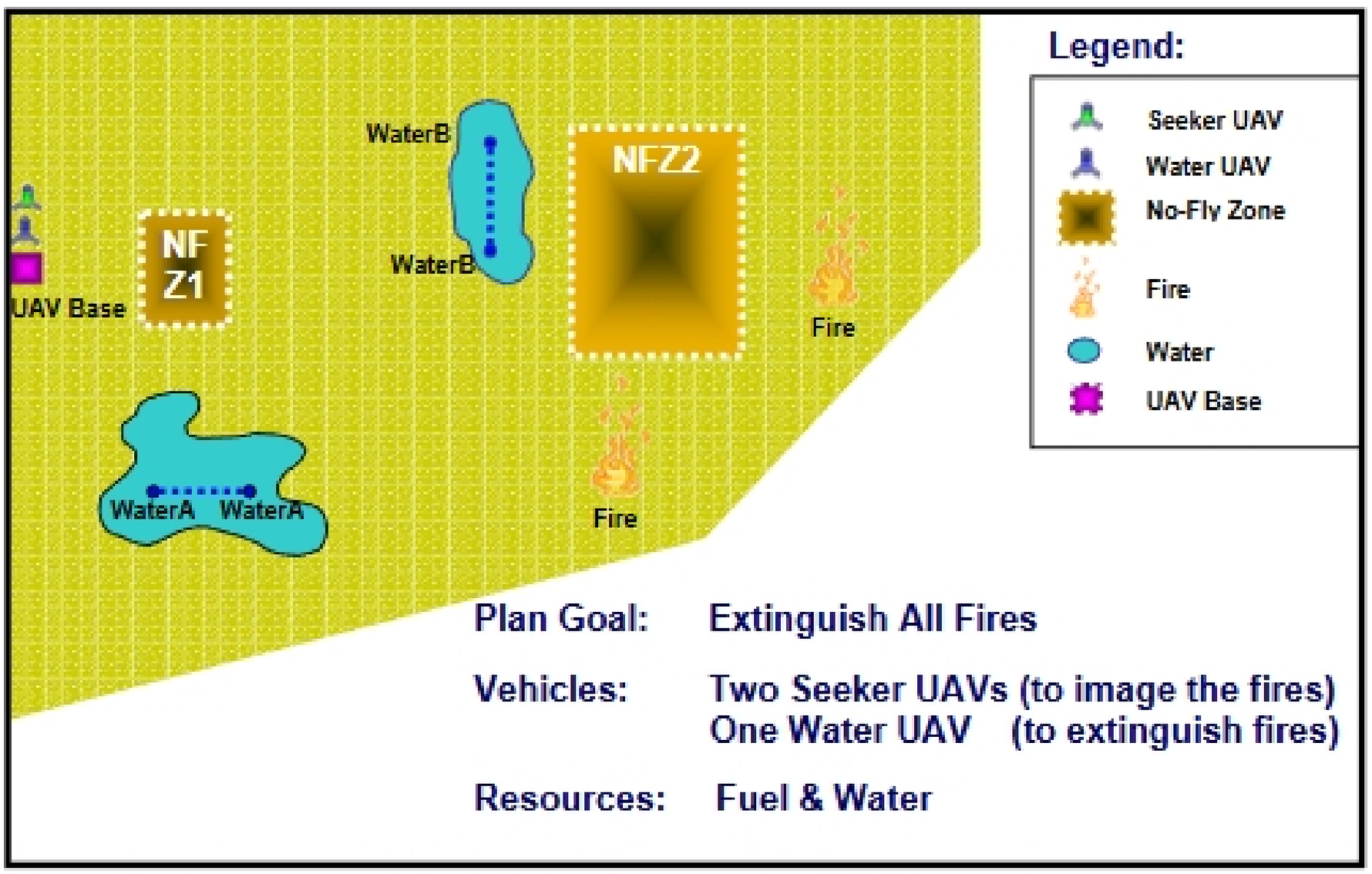

Figure 1: Autonomous Firefighting Cooperative UAV Scenario

To accomplish the goals set out above, the group has devised an integrated planner architecture that leverages the strengths of four separate planners. The four planners are Kirk, a strategic high-level mission planner, PDDL, a generative activity planner, dStarLite, a roadmap path-planner, and a MILP-based kinodynamic receding horizon path-planner. Kirk allows the operator to specify goals at an abstract level, in RMPL, The Reactive Model-Based Programming Language. The generative planner can then expand the high-level goals into lower-level activities. Then, the MILP-based receding horizon kinodynamic path planner generates an optimal motion plan, including obstacle avoidance. The dStarLite road-map path planner is called by all three afore-mentioned planners to get shortest-path distance to goal measurements and estimates.