Massachusetts Institute of Technology
16.412J/6.834J Cognitive Robotics
Project Proposal

Continuously Planning for Autonomous Navigation

Using

Conflict-Directed A*

to Generate

Temporal Flexible State Plans

Lawrence Bush, Brian Bairstow and Tony Jimenez
{ bushl2, bairstow}@mit.edu, tjimenez@alum.mit.edu

April 11, 2005

Massachusetts Institute of Technology
16.412J/6.834J Cognitive Robotics
Project Proposal

Lawrence Bush, Brian Bairstow and Tony Jimenez
{ bushl2, bairstow}@mit.edu, tjimenez@alum.mit.edu

April 11, 2005

Introduction:

Autonomous control of systems is an important topic, as autonomous systems can perform tasks that are dangerous, monotonous, or even impossible for humans. For example, Unmanned Aerial Vehicles (UAVs) can perform tasks such as reconnaissance, fire-fighting, and Mars exploration. Issues include the planning problem of where to go, and the control problem of how to actuate that movement. Bradley Hasegawa [1] posed the planning problem as a Selective Traveling Salesman Problem (STSP). Each point of interest (for example, a science site) is assigned a value, and each edge connecting points is assigned a cost. Then the most valuable and feasible ordered set of waypoints is chosen. Thomas Leaute [2] worked on the other end of the problem; starting from a plan and running the control system in a simulator. We hope to integrate the two techniques to get an autonomous planner and controller that can take a list of sites, decide on a plan, and then control the system to meet that plan. This will be a complete system that handles high and low level planning to achieve a goal.

Problem Statement:

Our project addresses the design of an autonomous exploratory planner for a UAV. This problem involves extending a continuous observation planning system [1] with an improved algorithm and merging this system with a kino-dynamic path planner [2].

Previous Work:

Our project is premised upon 3 main bodies of work, namely [1], [2] and [3]. The first step of our project involves integrating [1] and [3].

Continuous Observation Planning for Autonomous Exploration [1]

The thesis [1] presents a new approach for solving a robotic navigation path-planning problem. The approach first formulates the problem as a selective traveling salesman problem (S-TSP), then converts it to an optimal constraint satisfaction problem and solves it using the Constraint Based A* algorithm. The Solver, shown in the system architecture diagram in Figure 1, performs this key ability.

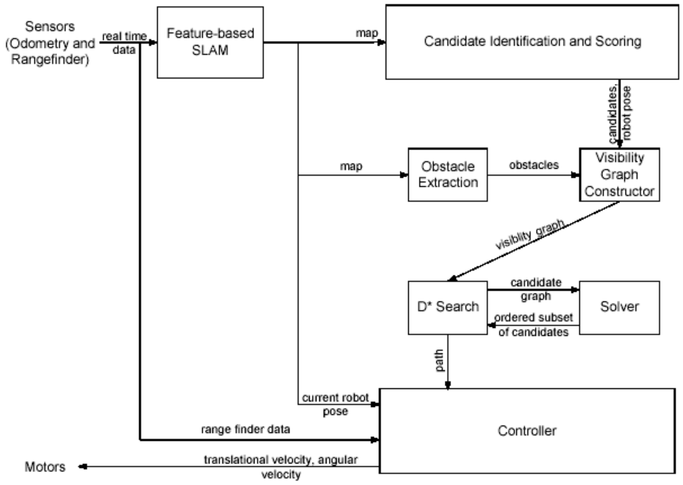

Figure 1: The above diagram is the system architecture for [1]. The navigation architecture starts with a partially complete map. Candidates and obstacles are extracted from the map, which are used to construct a visibility graph. The D* search is used to update the candidates. The candidates are passed to the solver, which creates a plan (ordered candidate subset).

The solver is a continuous observation planner, which updates the plan when new observations affect the candidate set (possible places to visit). The objective of the robot is to map its environment. The robot chooses to navigate to observation locations, which will maximize information gain. Each observation may affect the utility and cost of unvisited observation locations (candidates), which necessitates re-planning. There is an implicit trade-off between the planning horizon and how often the candidates are updated. The planning horizon should mirror the expected time period between re-planning. In other words, if we look ahead 5 tasks, we want to be able to execute those 5 tasks before we have to re-plan. If this does not occur, then our plan is optimized for a different planning period than it is executed for. This results in sub-optimal planning.

Ultimately, the system is making an exploration-exploitation trade-off, which can be generalized to other tasks. The tasks must involve observation and candidate list utility/cost updates. This method is likely to be effective when we have (at a minimum) a large-scale prior map of the exploration region.

The thesis [1] addressed a mapping application where the candidates frequently changed due to new observations. The finite horizon technique is more effective when the candidates do not change frequently. Yet the mapping application actually favors observation candidates that increase its situational knowledge the most. For these reasons, the finite horizon method is more effective when a high-level map is known. The attributes of continuous finite horizon planning lend themselves to exploratory missions with a specific objective (i.e. a science exploration application) where a prior map is known. Refining the map will affect the cost estimate for the science tasks and the utilities of the science tasks may change as prior successes affect the probability of future successes. This necessitates continuous planning. However, the changes should be sufficiently infrequent, so that a finite horizon is more effective that a purely greedy candidate selection strategy.

Key elements of the framework presented in [1] are shown in Table 1.

Exploration Problem:	Explore and construct a map of an environment
Exploration Method:	Feature based (Newman, Boss, and Leonard)
Assumption:	The robot knows the large-scale environment structure
Path Cost:	Path length (physical distance)
Path Planner:	Visibility Path Planner : F(map, candidates, pose)
Map Type:	Feature based SLAM map
Pose:	Robot position and heading
Candidate:	An observation point bordering an unexplored area
Candidate Utility:	An estimate of the observable unexplored area
Candidate Dynamics:	How do candidates change as a robot explores This is an open area of statistical learning research

Table 1: Key attributes of the continuous observation-planning framework.

Coordinating Agile Systems Through the Model-based Execution of Temporal Plans [2]

This work provides a novel model-based execution of a temporally flexible state plan for the purpose of UAV navigation. Its kino-dynamic controller is a continuous planning framework. However, the high level planner is not.

Our integration would enable this work to perform continuous high-level planning. The scope of our project includes enabling the simulation framework to accept TFSP updates. In particular, the observations would be considered when creating the temporally flexible state plan. Continuous high-level planning would allow the UAV to adapt its high-level goals to the observed environment. Specifically, it adapts the additional information that it learns about its environment (i.e. a more accurate map or the ramifications that one task has on the utility of future tasks). Therefore, this integration would provide a system, which observes, learns and updates its higher level planning goals.

Conflict-directed A* and Its Role in Model-based Embedded Systems [3]

This paper [3] introduces a method for solving optimal constraint satisfaction problems. An Optimal Constraint Satisfaction Problem (OCSP) is the problem of finding a consistent assignment of variables to values that is both consistent and has optimal value. More rigorously, such a problem is a 5-tuple $\langle x, y, D_x, C_x, g(y) \rangle$ where

- x is a set of variables each with domain D_x,
- C_x is a set of constraints on the variables that define what is a consistent assignment,
- $F(y)$ is a function $y \subset x \rightarrow \Re$ that defines the cost of an assignment.

Many problems can be formulated as OCSPs. Specifically, an S-TSP problem can be represented as an OCSP. CDA* has been shown to be an efficient implementation for solving OCSPs. For this reason, we proposed to replace the Constraint Based A* solver in [1] with a CDA* implementation.

Technical Approach:

Our first step will be to adapt Bradley Hasegawa's work (Continuous Observation Planning for Autonomous Exploration) [1] which involves a Selective Traveling Salesman Problem (STSP) with value updates. We have Conflict Directed A* (CDA*) code (in C++) from Tony Jimenez's work on the second assignment. We hope to gain access to the code for [1], and change it so that when it frames the STSP as an Optimal Constraint Satisfaction Problem (OCSP), it will use our CDA* algorithm to solve the OCSP. This may involve a lot of work in merging two separately developed algorithms.

Our next step will be to change the STSP solution to a Temporally Flexible State Plan (TFSP). This will involve adding time constraints to the ordered list of points to visit. This can be done trivially by generating the constraints at random or in a more logical manner by estimating times using a probabilistic motion model. We will probably use a simple C++ program to input the STSP solution and output a TFSP.

Our final goal is to link the work for [1] with the work by Thomas Leaute [2] on Coordinating Agile Systems Through The Model-based Execution of Temporal Plans. The work for [2] takes a TFSP and tests the controller using simulation. Conceivably we could start with an STSP, use the work for [1] with CDA* to solve it, turn the solution into a TFSP, and run the TFSP using Thomas Leaute's code in a simulator. However, [1] uses a STSP with changing values from observations, so it will keep producing different solutions over time. In discussion with Thomas Leaute, we learned that he thought it would be feasible to change the code for [2] to handle updates in the TFSP. Our fallback position would be to restart the simulation whenever the TFSP is updated.

As a final implementation though, we note that our motivating application is the Mars Airplane, where the expected value of future scientific tasks is affected by ongoing observations. The above integration of the continuous observation planner intends to allow plan updates due to new observations. These observations may affect the value of the future tasks. A science exploration mission lends itself to continuous (finite horizon) observation planning because it embodies an implicit exploration-exploitation trade-off. Our integration will, therefore, allow the simulator to extend the feedback-control loop to high level planning. This feedback-control loop integration, however, is left as future work.

Plan:

Our minimal plan will be an extension to [1]. To accomplish our learning goals, our objective will be to master the concepts presented in Bradley Hasegawa's thesis and develop an extension to his work by replacing constraint-based A* with conflict-directed A*.

Our baseline plan would be a further extension to the above-mentioned algorithm. We will then take the algorithm developed by Mr. Hasegawa and convert the solution to the Selective Traveling Salesman Problem (STSP) into a Temporally Flexible State Plan (TFSP).

Our enhanced plan is a novel cognitive robot application. This plan would be to take the extension to the previous work from [1] and merge it with [2] by modifying it to accept a continuously updated TFSP. The resulting application would be able to take as inputs a set of waypoints and continuously plan a kino-dynamic path that will be optimized according to the utility of the waypoints. This cognitive robot application can then be executed on a hardware-in-the-loop simulation as cited in [2] if that equipment is available from the MERS lab.

Schedule:

4/11	Turn in proposal
4/17	Integrate [1] with CDA*
4/19	Create translation from STSP solution to TFSP. Reassess situation and schedule
4/24	Integrate [1] with [2], modify [2] to handle updates to TFSP. Coding complete
4/27	Simulations complete
5/4	Paper complete
5/7	Presentation complete

Division of Labor:

- Adapt Bradley Hasegawa's work
  - ❑ Solve OCSP using CDA*
  - ❑ Convert OCSP [1] representation to CDA* [2] representation
- Change the STSP solution to a TFSP
  - ❑ add time constraints
- Link Bradley Hasegawa's work with the work by Thomas Leaute
  - ❑ Adapt his code so that the TFSP can be continuously replaced with a new one.
- Overarching Activities:
  - ❑ Debugging
  - ❑ Interfacing
    - ○ Interfacing can often result in unforeseen implementation delays. Therefore, we intend to address most interfacing issues upfront, which will allow us to properly adapt the scope of our project.
    - ○ We intend to do this by first implementing the interface between the two algorithms ([1] and [2]) without observations. Our second step will involve properly formatting the TFSP for the simulation interface [2]. The simulation framework [2] will initially require no fundamental changes since the TFSP will be static. The second spiral of our project will then include observations and TFSP updates. This will require a fundamental change to the simulation framework and interface [2].
  - ❑ Write-up

References:

[1] B. Hasegawa, "M.Eng. Thesis: Continuous Observation Planning for Autonomous Exploration," Massachusetts Institute of Technology,(2004). http://mers.csail.mit.edu/theses/BradHasegawaThesis.pdf

[2] T. Leaute and B. Williams, "Coordinating Agile Systems Through the Model-based Execution of Temporal Plans," accepted to the *International Workshop on Planning under Uncertainty for Autonomous Systems* (2005).

[3] B. Williams and R. Ragno, "Conflict-directed A* and Its Role in Model-based Embedded Systems," to appear in the *Special Issue on Theory and Applications of Satisfiability Testing*, accepted in *Journal of Discrete Applied Math* (January 2003). http://mers.csail.mit.edu/papers/jdam.pdf