Your Title Goes Here (It Can Be Really Really Really Really Long)

Your Name Here

Abstract

A nice abstract goes here.

Acknowledgments

Some acknowledgments go here.

Your Title Goes Here (It Can Be Really Really Really Really Long)

Your Name Here

A departmental senior thesis submitted to the
Department of Computer Science at Trinity University
in partial fulfillment of the requirements for graduation
with departmental honors.

April 1, 2005

Thesis Advisor	Department Chair

Associate Vice President
for
Academic Affairs

Your Title Goes Here (It Can Be Really Really Really Really Long)

Your Name Here

Contents

List of Tables

List of Figures

Chapter 1

Example chapter

Example chapter, with apologies to Alex Kolliopoulos, from whose thesis the examples of math and tables were borrowed.

1.1 Examples of figures and tables

This section contains some words, plus Figure 1.1 and Table 1.1.

words, words,

words, words,

1.2 Examples of math

This section contains some math. First, here's a set of equations.

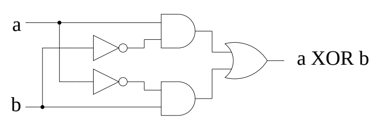

Figure 1.1: An example figure.

Trial 1	Expanding a node to select a child
Trial 3	Selecting a node near the middle of a long, linear list
Trial 4	Selecting a node near the top of a long, linear list
Trial 5	Selecting a node near the bottom of a long, linear list
Trial 6	Scrolling and expanding folders in a large tree
Trial 7	Finding a node deep and near the bottom in a large tree
Trial 8	Finding a node near the top of a large tree

Table 1.1: Purposes of each experimental trial.

$$
\begin{aligned}
y_p &= \frac{y}{\sqrt{y^2+a^2}}, \\
y_p^2 &= \frac{y^2}{y^2+a^2}, \\
y_p^2 &= \frac{y^2+a^2-a^2}{y^2+a^2}, \\
y_p^2 &= 1-\frac{a^2}{y^2+a^2}, \\
y_p^2-1 &= -\frac{a^2}{y^2+a^2}, \\
1-y_p^2 &= \frac{a^2}{y^2+a^2}.
\end{aligned}
$$

words, words,

Now here's a numbered equation.

$$0 = 0 \tag{1.1}$$

1.3 Examples of references

Section 1.1 contains Figure 1.1 and Table 1.1. Section 1.2 contains Equation (1.1). The sample bibliography file contains references to a book [**?**] and a Web site [**?**], plus some other things.

Chapter 2

Partially Observable Environments

2.1 Introduction

Until now, both the GTGR and GTGRD models have given the observer full knowledge of the adversary's state for the entirety of the game. In real-world environments, observers may not have perfect information regarding the states and actions of an adversary.

To accommodate for scenarios with incomplete information for the adversary, we introduce a partially observable variant of the GTGR scenario. In partially observable scenarios, the rules of the game remain largely unchanged, except for addition of shadow states." The observer can not discern the current state of the adversary, while the adversary occupies a shadow state. When the adversary enters an observable portion of the graph, the observer will become aware of the adversary's position once more.

Figure 2.1 illustrates a partially observable environment. Visible states, in which the observer can see the adversary white. Shadow states, in which the adversary is hidden from the observer, are black. The agent starts the game in state S. When the adversary moves to states 4 ,5, 6, or 7, the observer is unable to determine their position until the adversary

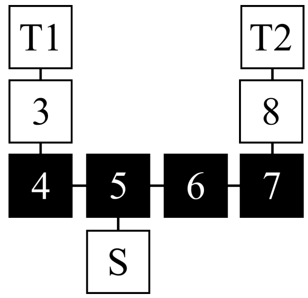

Figure 2.1: A partially observable graph.

re-enters a visible portion of the graph. We will examine two solutions to the partially observable model, both of which involving linear programming.

2.2 The Whale Method

The first method of solving partially observable environments, which we will call the "Whale Method," will utilize disjoint sets of shadow states. We call these sets of shadow states "shadow sets." We say that two shadow states belong to the same shadow set, if the adversary can travel between the two without entering an observable state.

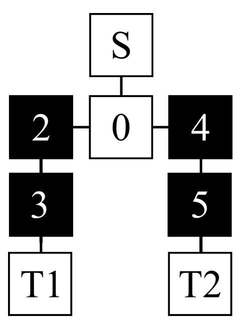

Figure 2.2: A partially observable graph with two shadow sets.

The graph in Figure 2.2 has two disjoint shadow sets, one composed of states 2 and 3, the other composed of states 4 and 5. In using the Whale Method, the observer treats each shadow set a single state, which we will call a "whale state." We can identify the single shadow set in Figure 2.1, composed of states 4, 5, 6, and 7. In using the Whale method, the observer treats each state in the shadow set as the same state. While the adversary may require several turns to travel among states 4, 5, 6, and 7, the adversary will act as if the has decided to remain stationary in the newly created state W seen in Figure 2.3

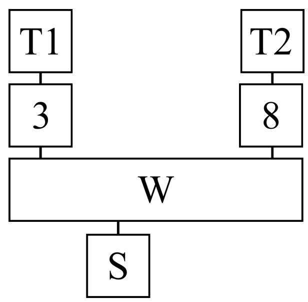

Figure 2.3: A partially observable graph with a single state in place of shadow states.

We can add the following to the mixed integer program to accommodate for partially observable environments when using the Whale method.

$$V(\theta, s) \leq \sum_{i \in B} r(s, i, j, \theta) f_i(s) + V(\theta, j) \forall \theta \in B, \forall s \mid s \neq \theta, s \notin H, \forall j \in \nu(s) \tag{3}$$

$$V(\theta, s) \leq \sum_{i \in B} r(s, i, j, \theta) f_w(s) + V(\theta, j) \forall \theta \in B, \forall s \mid s \neq \theta, s \in H, \forall j \in \nu(s) \tag{3}$$

$$\sum_{i} f_i(s) = 1 \quad \forall s \tag{3}$$

$$\sum_{w} f_w(s) = 1 \quad \forall s \tag{3}$$

$$f_i(s) \geq 0 \quad \forall s, i \tag{2.1}$$

$$f_w(s) \geq 0 \quad \forall s, w \tag{2.2}$$

We let H denote the set of all shadow states, and w denote the whale state the observer knows the adversary to be occupying. An observer action for any whale state w is written as $f_w(s)$. These changes to the linear program require the observer to take the same action for each turn the adversary spends in a particular shadow set. The performance of the Whale method will be examined in a later section.

2.3 Transmogrification Method

Appendix A

Example appendix

Here is my code.

```
#include <iostream>
int main(void) {
    cout << "Hello, world!\n";
    return 0;
}
```