Sokoban: Search in a complex domain

Yann Chazallon, Nicolas Dossou-Gbété, Tony Chan Ki Hong
and Michal Staniaszek

October 19, 2013

Table of Contents

Problem: Sokoban

- A puzzle game which written in 1981 by Hiroyuki Imabayashi
- Control an agent to push boxes onto goal locations
- PSPACE-complete single agent search problem

Why is it interesting

- It can be applied in real-life situation
- High Branching factor: 4N where N = number of box
- Depth of the search tree can be infinity

Table of Contents

Map Representation

- Wall: # Player: @ Player in goal: +
- Goal: . Box: $ Box in goal: *
- Recieve map as a txt file with symbols representing different objects
- Static Objects in Static Board
- Dynamic Objects + Static Objects in Board

Deadlock Detection

- Player cannot pull a box, may push box into lose state
- Static Lock : push the box to the goal even without existance of other boxes
- Dynamic Lock : some boxes blocking in the middle and impossible to remove the blocking box

Search Method

- A*
- Best First
- BFS

Heuristic Method

- first item
- second item

1. list one
2. list two

Table of Contents

1. Introduction
2. Methods
3. Implementation
4. Evaluation
5. Discussion

Object-Oriented Programming

- Board class - Stores the state of the board and methods to change Board State
- Search class - Allow bi-directional search
- SearchNode class
-

Table of Contents

- first item
- second item

1. list one
2. list two

Table of Contents

1. Introduction
2. Methods
3. Implementation
4. Evaluation
5. Discussion

Things that we can do better

- Members may not fully understand the code
- Didn’t meet too often
- Only have a vague idea

Initial Idea

- Use motion of the player as the state expansion
- A Search Implementation

Current Method

- More efficent equality check
- improved heuristic - minimum matching heuristic

What will we do differently if we do again

- Start with simple implementation that worked
- A better heuristic can be used