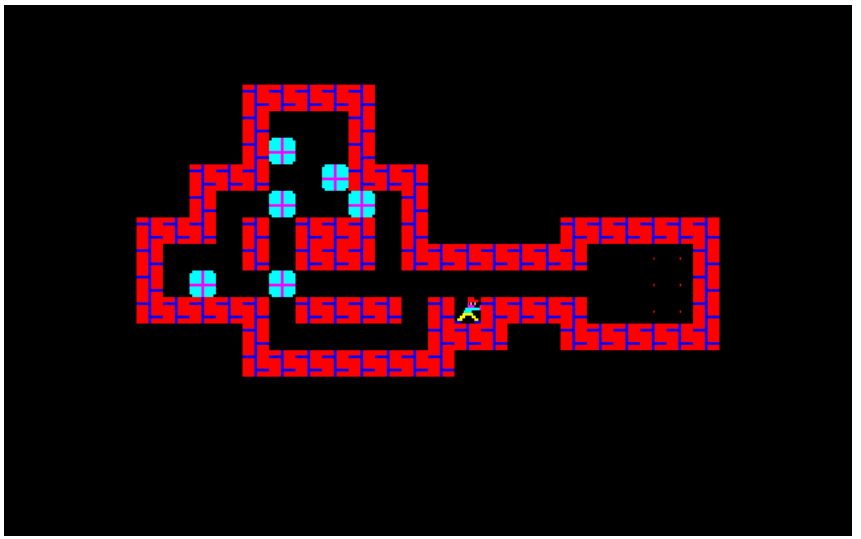
Sokoban Solver

A Push in the Right Direction

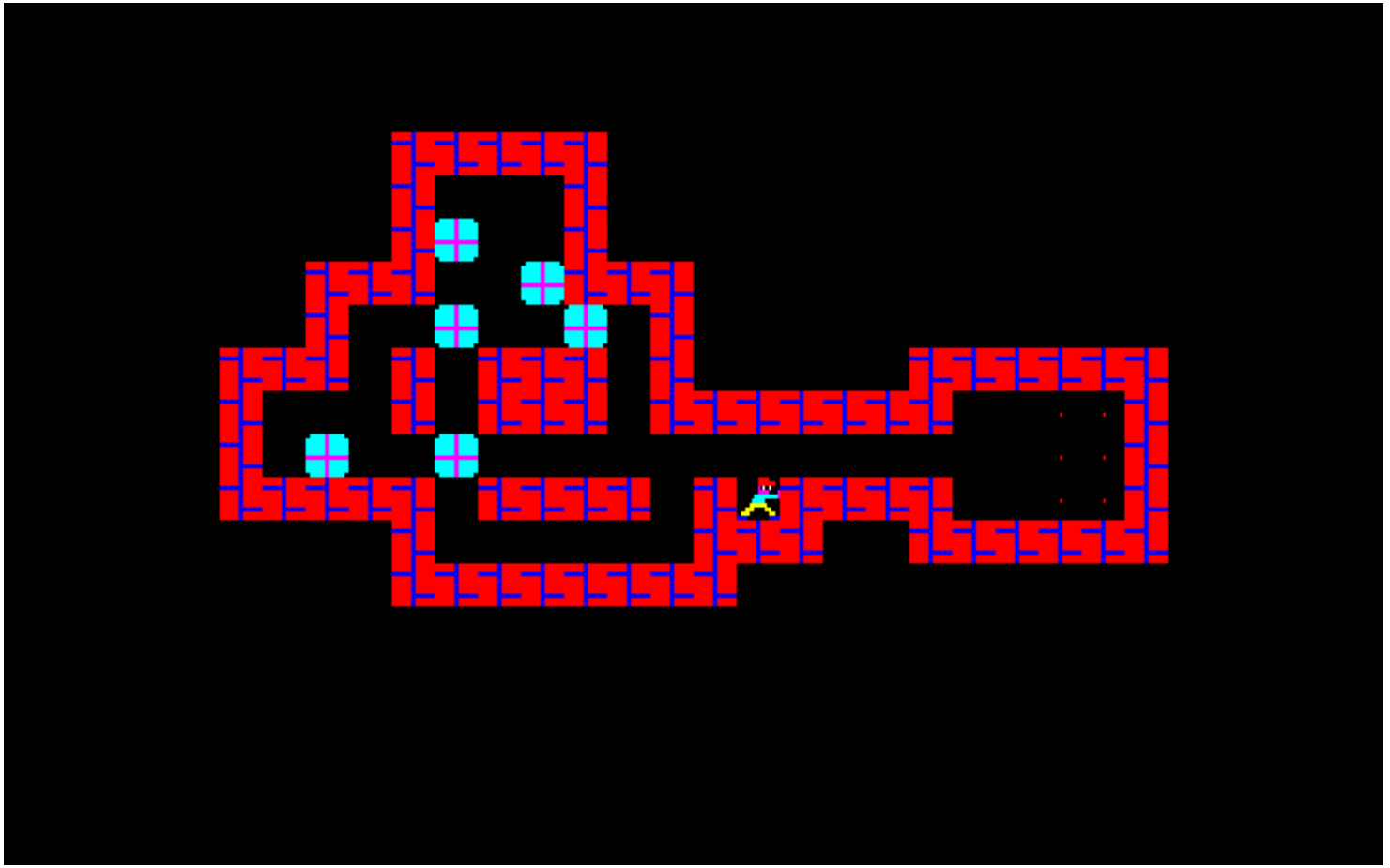
Simon Kotlinski, Rasmus Göransson, Peter Boström

Sokoban Solver

- About Sokoban
- Problem
- Method
- Optmizations
  - Repeated states
  - Prioritizing states
  - Assign State Cost
  - Deadlocks
  - Converting immovable boxes into walls
  - Partial bipartite matching
  - Compiler optimizations
  - Tunnel warping
- Result and conclusions



About Sokoban



By Hiroyuki Imabayashi in 1981.

Problem

Level element	Character	ASCII Code
Wall	#	0x23
Player	@	0x40
Player on goal square	+	0x2b
Box	$	0x24
Box on goal square	*	0x2a
Goal square	.	0x2e
Floor	(Space)	0x20

Method

- Fetch board from server and create the level.
- Create the first board state.
- Create new states from the best state.
- Repeat until we find a solution.
- Backtrack a solution string.
- Send the solution string to the server.

Optimizations

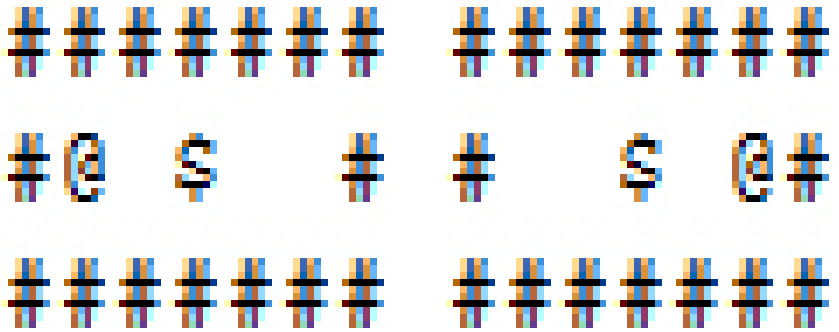
- Repeated states
- Prioritizing states
- Assign State Cost
- Deadlocks
- Converting boxes into walls
- Partial bipartite matching
- Compiler optimizations
- Tunnel warping

Repeated states

- Lookups and insertions in hash set is fast.
- Hash key from a set of box-positions.
- Sort boxes before hashing.

  (Permutations are equal.)
- Player position does not effect hash-value. (Intentionally)
- Finally, states are equal when boxes are equally placed and the player from one state can *reach* the player position in the other state.

Repeated states



Two different states with the same hash-value.

Assigning state costs

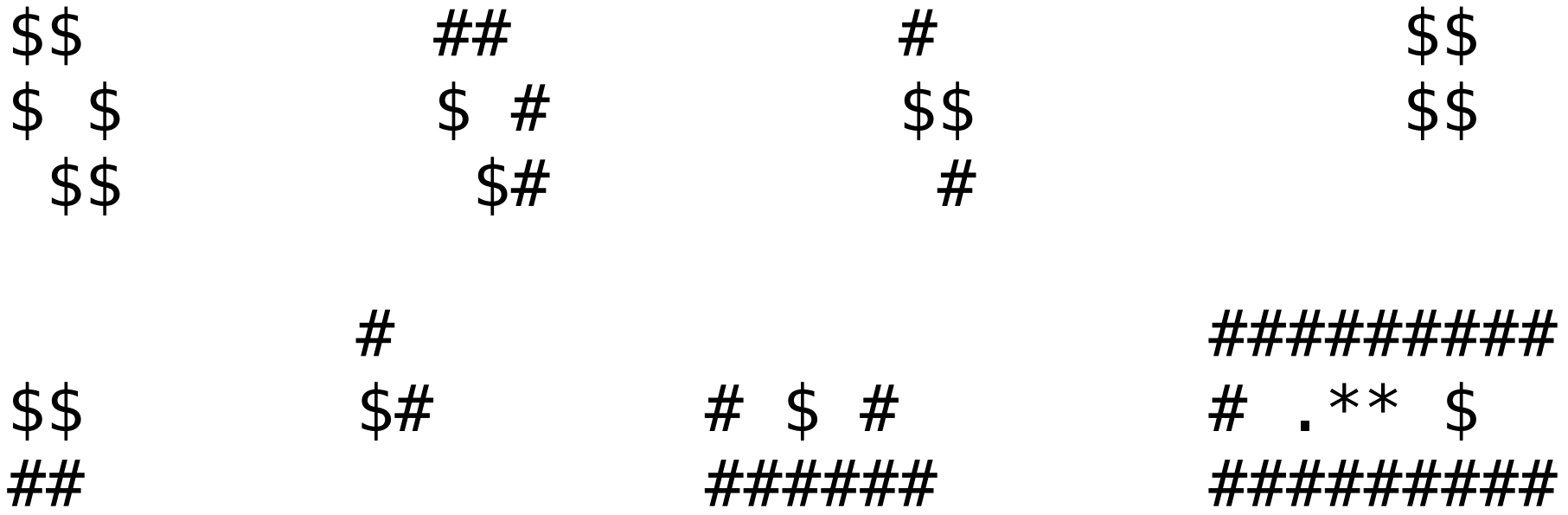
- Number of pushes from a goal.
- 100 plus for every box not on a goal.
- Backward breadth-first search.(Pull boxes)

Assigning state costs

```
##########
#-210123-#
##########
```

The 0 is a goal and 1,2,3 are the distance to the goal. The dashes are dead positions which cause deadlocks if any box is moved here.

Deadlocks



Examples of deadlocks. States from which no solution can be found.

Future Deadlock Work

- Corral deadlocks.

```
##########
#. @   $ #
#.     $ #
##########
```

A corral deadlock; the player cannot reach the area to the right without causing a deadlock. The boxes are dead, and the state is therefore also dead.

Tunnel warping

- Only one box in a tunnel at most.
- Implemented by discarding partial pushes in tunnels.
- When a box enters a tunnel, it's pushed, either to a goal or to the end of the tunnel.

Converting Boxes into Walls

- Converting immovable boxes on goals into walls.
- Finds deadlocks where other boxes cannot find any goal anymore.
- Board #35 went from hundred thousand expands into less than 100.

```
#######     #######
  .*$       #.** $
            #.** $
            ##
```

Partial bipartite matching

- Finds deadlocks where a goal can not be reached by any box.

```
##########
# . $ $  #
# .    @ #
#        #
##########  “Oops!”
```

Compiler Optimizations

- ”-O3” flag in g++
- More than twice the time to search.
- Lets us pass the boards in the set which our solver finds the hardest

Result and Conclusions

- 136 test boards in 4 minute and 21 seconds.
- Compiler optimizations: 19 minutes to 7 minutes.
- Immovable boxes to walls: 100 000 expands to less than 100 on map 35.
- Different maps have different deadlocks.
- The hash set is important for avoiding loops.
- The priority-queue is fundamental for the whole program.