Foundations of Artificial Intelligence

B3. State-Space Search: Examples of State Spaces

Malte Helmert

University of Basel

March 4, 2026

State-Space Search: Overview

Chapter overview: state-space search

- ▶ B1–B3. Foundations
    - ▶ B1. State Spaces
    - ▶ B2. Representation of State Spaces
    - ▶ B3. Examples of State Spaces
- ▶ B4–B8. Basic Algorithms
- ▶ B9–B15. Heuristic Algorithms

Three Examples

In this chapter we introduce three state spaces
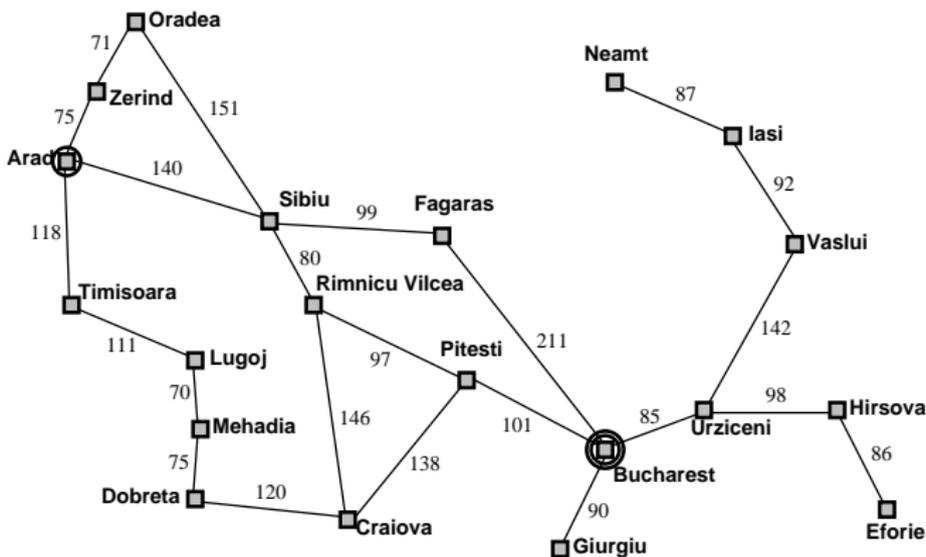that we will use as illustrating examples:

1. route planning in Romania
2. blocks world
3. missionaries and cannibals

B3.1 Route Planning in Romania

Route Planning in Romania

Setting: Route Planning in Romania
We are on holiday in Romania and are currently located in Arad.
Our flight home leaves from Bucharest. How to get there?

Romania Formally

> ### State Space Route Planning in Romania
> ▶ states S: {arad, bucharest, craiova, ..., zerind}
> ▶ actions A: $move_{c,c'}$ for any two cities c and c'
>   connected by a single road segment
> ▶ action costs $cost$: see figure,
>   e.g., $cost(move_{iasi,vaslui}) = 92$
> ▶ transitions T: $s \xrightarrow{a} s'$ iff $a = move_{s,s'}$
> ▶ initial state: $s_I = $ arad
> ▶ goal states: $S_G = $ {bucharest}

B3.2 Blocks World

Blocks World
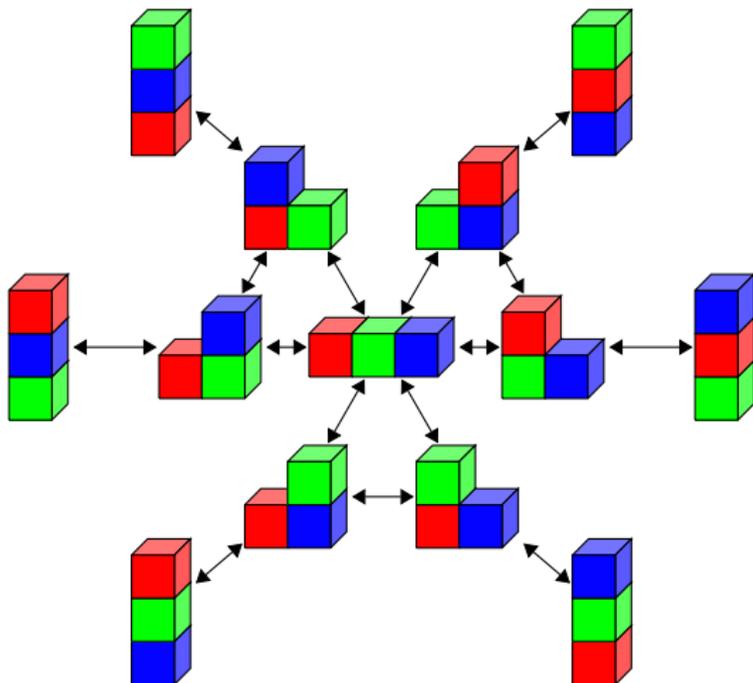
Blocks world is a traditional example problem in AI.

Setting: Blocks World

▶ Colored blocks lie on a table.

▶ They can be stacked into towers, moving one block at a time.

▶ Our task is to create a given goal configuration.

Example: Blocks World with Three Blocks

Action names omitted for readability. All actions cost 1.

Initial state and goal can be arbitrary.

Blocks World: Formal Definition

state space $\langle S, A, cost, T, s_I, S_G \rangle$ for blocks world with n blocks

> ### State Space Blocks World
>
> states S:
>
> partitions of $\{1, 2, \ldots, n\}$ into nonempty ordered lists
>
> example $n = 3$:
>
> ► $\{\langle 1, 2, 3 \rangle\}$, $\{\langle 1, 3, 2 \rangle\}$, $\{\langle 2, 1, 3 \rangle\}$,
> $\{\langle 2, 3, 1 \rangle\}$, $\{\langle 3, 1, 2 \rangle\}$, $\{\langle 3, 2, 1 \rangle\}$
>
> ► $\{\langle 1, 2 \rangle, \langle 3 \rangle\}$, $\{\langle 2, 1 \rangle, \langle 3 \rangle\}$, $\{\langle 1, 3 \rangle, \langle 2 \rangle\}$,
> $\{\langle 3, 1 \rangle, \langle 2 \rangle\}$, $\{\langle 2, 3 \rangle, \langle 1 \rangle\}$, $\{\langle 3, 2 \rangle, \langle 1 \rangle\}$
>
> ► $\{\langle 1 \rangle, \langle 2 \rangle, \langle 3 \rangle\}$

Blocks World: Formal Definition

state space $\langle S, A, cost, T, s_I, S_G \rangle$ for blocks world with n blocks

---

State Space Blocks World

actions A:

- $\{move_{u,v} \mid u, v \in \{1, \ldots, n\}$ with $u \neq v\}$
    - move block u onto block v.
    - both must be uppermost blocks in their towers
- $\{to\text{-}table_u \mid u \in \{1, \ldots, n\}\}$
    - move block u onto the table ($\rightsquigarrow$ forming a new tower)
    - must be uppermost block in its tower

action costs $cost$:

$cost(a) = 1$ for all actions $a \in A$

Blocks World: Formal Definition

state space $\langle S, A, cost, T, s_I, S_G \rangle$ for blocks world with n blocks

---

State Space Blocks World
transitions:

▶ transition $s \xrightarrow{a} s'$ with $a = move_{u,v}$ exists iff
  ▶ $s = \{\langle b_1, \ldots, b_k, u \rangle, \langle c_1, \ldots, c_m, v \rangle\} \cup X$ and
  ▶ if $k > 0$: $s' = \{\langle b_1, \ldots, b_k \rangle, \langle c_1, \ldots, c_m, v, u \rangle\} \cup X$
  ▶ if $k = 0$: $s' = \{\langle c_1, \ldots, c_m, v, u \rangle\} \cup X$

▶ transition $s \xrightarrow{a} s'$ with $a = to\text{-}table_u$ exists iff
  ▶ $s = \{\langle b_1, \ldots, b_k, u \rangle\} \cup X$ with $k > 0$ and
  ▶ $s' = \{\langle b_1, \ldots, b_k \rangle, \langle u \rangle\} \cup X$

Blocks World: Formal Definition

state space $\langle S, A, cost, T, s_I, S_G \rangle$ for blocks world with n blocks

---

State Space Blocks World

initial state s_I and goal states S_G:

one possible scenario for $n = 3$:

► $s_I = \{\langle 1, 3 \rangle, \langle 2 \rangle\}$

► $S_G = \{\{\langle 3, 2, 1 \rangle\}\}$

(in general can have arbitrary scenarios)

Blocks World: Properties

| blocks	states	blocks	states
1	1	10	58941091
2	3	11	824073141
3	13	12	12470162233
4	73	13	202976401213
5	501	14	3535017524403
6	4051	15	65573803186921
7	37633	16	1290434218669921
8	394353	17	26846616451246353
9	4596553	18	588633468315403843

▶ For every given initial and goal state with n blocks, simple algorithms find a solution in time $O(n)$. (How?)

▶ Finding optimal solutions is NP-complete (with a compact problem description).

B3.3 Missionaries and Cannibals

Missionaries and Cannibals

Setting: Missionaries and Cannibals

- ▶ Six people must cross a river.
- ▶ Their rowing boat can carry one or two people across the river at a time.
  (It is too small for three.)
- ▶ Three people are missionaries, three are cannibals.
- ▶ Missionaries may never stay with a majority of cannibals.
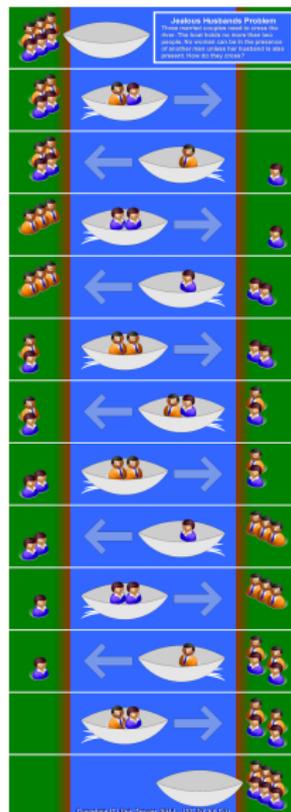
Missionaries and Cannibals Formally

State Space Missionaries and Cannibals

states S:

triples of numbers $\langle m, c, b \rangle \in \{0, 1, 2, 3\} \times \{0, 1, 2, 3\} \times \{0, 1\}$:

- ▶ number of missionaries m,
- ▶ cannibals c and
- ▶ boats b

on the left river bank

initial state: $s_I = \langle 3, 3, 1 \rangle$

goal: $S_G = \{\langle 0, 0, 0 \rangle, \langle 0, 0, 1 \rangle\}$

actions, action costs, transitions: ?

B3.4 Summary

Summary

illustrating examples for state spaces:

- ▶ route planning in Romania:
  - ▶ small example of explicitly representable state space
- ▶ blocks world:
  - ▶ family of tasks where n blocks on a table must be rearranged
  - ▶ traditional example problem in AI
  - ▶ number of states explodes quickly as n grows
- ▶ missionaries and cannibals:
  - ▶ traditional brain teaser with small state space
    (32 states, of which many unreachable)