

a. 目标: 从河岸A 将 3个传教士和3个野人运到河岸B.

状态抽象: 河岸A 当前 传教士数目 c , 野人数目 m , 是否有船 b

用三元组 $[c, m, b]$ 表示.

初始状态: $[3, 3, 1]$

目标状态: $[0, 0, 0]$

约束: $c \geq m$ 且 $3-c \geq 3-m \Leftrightarrow c = m$.

为 $m=0, c=1, 2, 3$

或 $c=0, m=1, 2, 3$

后继函数: $b=1 \rightarrow b=0$ 且 $\begin{cases} a=a-1, c=c-1 \\ a=a-2 \\ c=c-2 \\ a=a-1 \\ c=c-1 \end{cases}$

$b=0 \rightarrow b=1$ 且 $\begin{cases} a=a+1, c=c+1 \\ a=a+2 \\ c=c+2 \\ a=a+1 \\ c=c+1 \end{cases}$

且满足约束

完整状态空间 $[3, 3, 1], [3, 2, 1/0], [3, 1, 1/0], [3, 0, 1/0], [2, 2, 1/0], [1, 1, 1/0], [0, 3, 1/0], [0, 2, 1/0], [0, 1, 1/0], [0, 0, 0]$ 共 18 种状态

b. 搜索算法: DFS 策略.

从初态 $[3, 3, 1]$ 开始, 在操作后, 检查是否为状态空间中某一个.

如果有, 改变状态, 检查新状态, 继续操作, 直到处于 $[0, 0, 0]$ 状态.

如果没有, 则 回退到前一步状态 (深搜停止).

需检查重复状态, 防止出现死循环.

c. 状态空间简单, 但反复 检测重复状态是困难的.