

# Homework 2

PB17000297 罗晏宸

March 8 2020

## 1 EXERCISE 3.6

跟踪 A\* 算法应用直线距离启发式求解从 Lugoj 到 Bucharest 问题的过程。给出结点扩展的顺序和每个结点的  $f$ 、 $g$  和  $h$  值。

解 Romania 问题中到 Bucharest 的直线距离如表所示：

<b>Arad</b>	366	<b>Mehadia</b>	241
<b>Bucharest</b>	0	<b>Neamt</b>	234
<b>Craiova</b>	160	<b>Oradea</b>	380
<b>Drobeta</b>	242	<b>Pitesti</b>	100
<b>Eforie</b>	161	<b>Rimnicu Vilcea</b>	193
<b>Fagaras</b>	176	<b>Sibiu</b>	253
<b>Giurgiu</b>	77	<b>Timisoara</b>	329
<b>Hirsova</b>	151	<b>Urziceni</b>	80
<b>Iasi</b>	226	<b>Vaslui</b>	199
<b>Lugoj</b>	244	<b>Zerind</b>	374

表 1:  $h_{SLD}$  的值——到 Bucharest 的直线距离

使用直线距离启发式，应用 A\* 算法，过程如下：

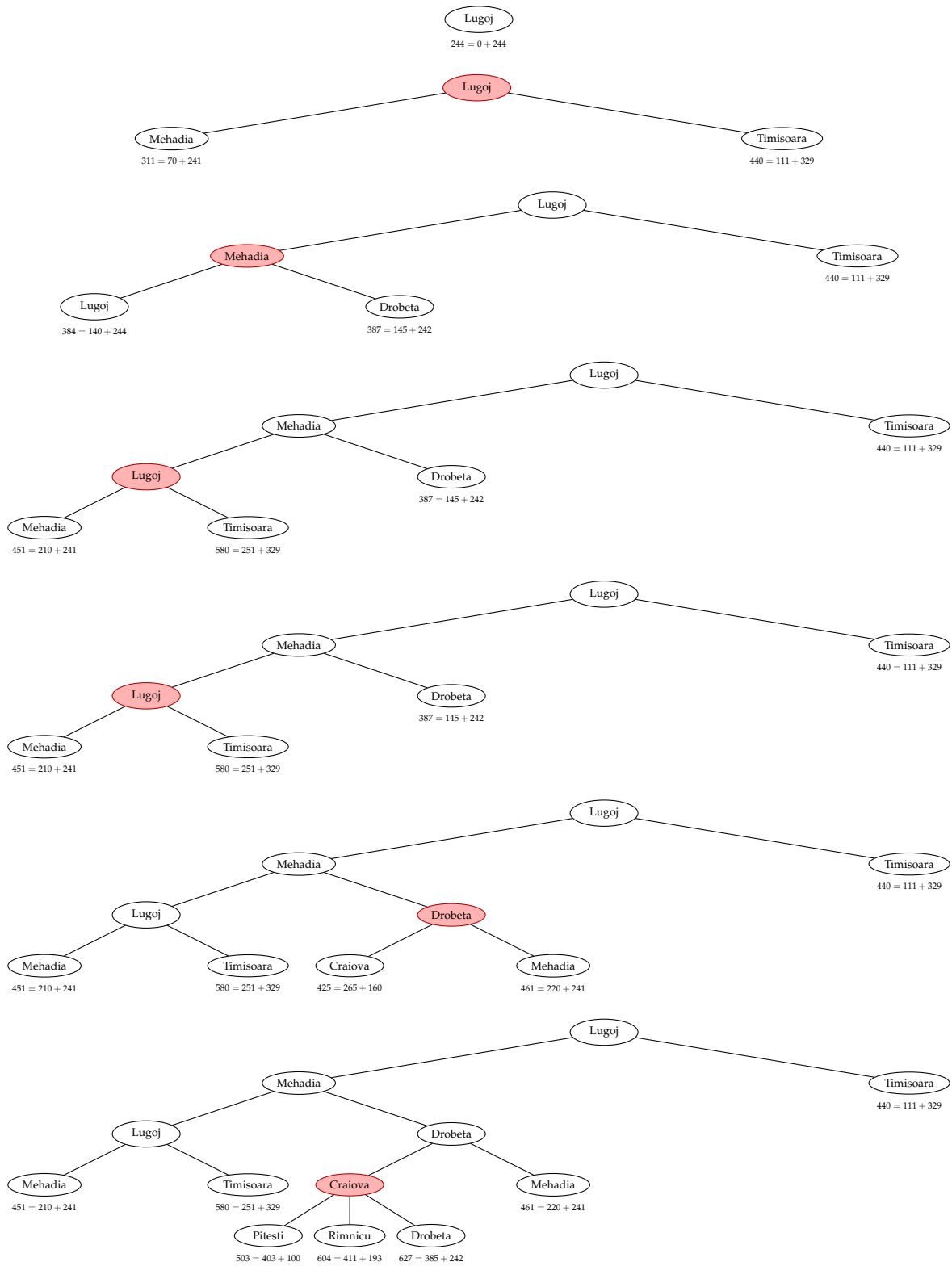


图 1: 使用 A\* 搜索应用直线距离启发式求解从 Lugoj 到 Bucharest 问题



## 2 EXERCISE 3.25

启发式路径算法(Pohl,1977)是一种最佳优先搜索,它的评估函数是  $f(n) = (2-w)g(n) + wh(n)$ , 假设  $h$  是可采纳的。 $w$  取什么值能保证算法是最优的? 当  $w = 0$ ,  $w = 1$ ,  $w = 2$  时, 分别是什么搜索算法?

解 由于  $h$  是可采纳的, 为了保证一致性, 应有:

$$\begin{cases} 2-w > 0 \\ w \geq 0 \end{cases}$$

同时成立, 即  $0 \leq w < 2$  时算法是最优的。

当  $w = 0$  时,  $f(n) = 2g(n)$ , 算法退化为 (带有常系数的) 一致代价搜索; 当  $w = 1$  时,  $f(n) = g(n) + h(n)$ , 算法即  $A^*$  搜索; 当  $w = 2$  时, 算法是 (带有常系数的) 贪婪最佳优先搜索。

## 3 EXERCISE 3.28

设计一个启发函数, 它在八数码问题中有时会估计过高, 并指出对某一特定问题它会求出次优解 (可以用计算机编程找出)。证明: 如果  $h$  被高估的部分不超过  $c$ ,  $A^*$  算法返回的解代价比最优解代价多出的部分也不超过  $c$ 。

解 启发式函数为不在位的棋子数与所有棋子到其目标位置的曼哈顿距离和之和, 即  $h = h_1 + h_2$ 。对于如图3的状态 其启发式函数为  $h = 1 + 1 = 2$ , 高于其实际的解代价 1。

1		2
3	4	5
6	7	8

图 3: 一个在八数码问题中估计过高的实例

证明. 设  $h$  被高估的部分不超过  $c$ , 即  $h(n) \leq h^*(n) + c$ , 则对于最优解路径上的任意结点  $n$ ,

有

$$\begin{aligned}f(n) &= g(n) + h(n) \\ &\leq g(n) + h^*(n) + c \\ &\leq C^* + c\end{aligned}$$

即 A\* 算法返回的解代价比最优解代价多出的部分也不超过  $c$ 。

□

## 4 EXERCISE 3.29

证明如果启发式是一致的，它一定是可采纳的。构造一个非一致的可采纳启发式。

**解** 设启发式  $h(n)$  是一致的，即对于每个结点  $n$  和通过任一行动  $a$  生成的  $n$  的每个后继结点  $n'$ ，从结点  $n$  到达目标的估计代价不大于从  $n$  到  $n'$  的单步代价与从  $n'$  到达目标的估计代价之和：

$$h(n) \leq c(n, a, n') + h(n')$$

**证明.** 对从  $n$  到任何目标的最短路径上的结点数  $k$  进行归纳：

当  $k = 1$  时，取目标结点即  $n'$ ，则  $h(n) \leq c(n, a, n') + h(n')$  成立。

当  $k \leq 1$  时，假设  $h(n')$  是可采纳的，则有

$$h(n) \leq c(n, a, n') + h(n') \leq c(n, a, n') + h^*(n') = h^*(n)$$

即  $h(n)$  对  $k + 1$  仍是可采纳的。

综上， $h(n)$  是可采纳的。

□

在 Romania 问题中，如果使用到达 Bucharest 的最少经过一个中间结点的路径距离减去到达后继结点的最大直线距离作为启发式，这个启发式是非一致的，它并不满足三角不等式，但它仍不会给出过高的估计，因此是可采纳的。

## 5 EXERCISE 6.5

使用带前向检验、MRV 和最少约束值启发式的回溯算法手动求解图中的密码算术问题

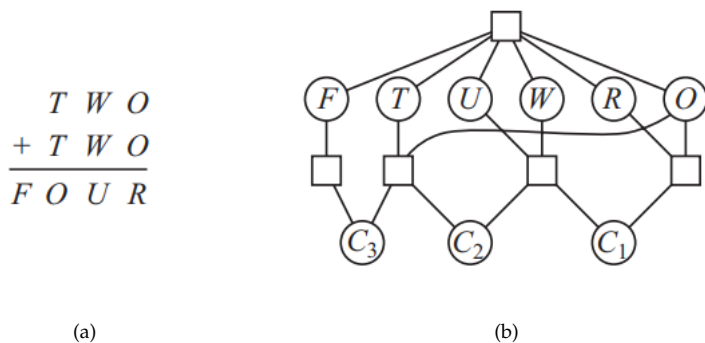


图 4:

(a) 密码算术问题。不同字母表示不同的数字；目标是找到能够使加法算式成立的代替字母的数字，附加约束是最前面的数字不能是 0。(b) 密码算术的约束超图，给出的是 *AllDiff* 约束（最上方的方框）和每列的相加约束（中间的四个方框）。变量  $C_1$ 、 $C_2$ 、 $C_3$  表示每列进位

解 手动求解过程如下：

- i 选择变量  $C_3$ ，其表示进位，因此有  $C_3 \in \{0, 1\}$
- ii 取  $C_3 = 1$ ，由前向检验限制，附加约束是最前面的数字不能是 0，因此  $F$  不能取 0，故  $C_3$  不能取 0
- iii 选择变量  $F$ ，其只有一个可能的值
- iv 取  $F = 1$
- v 选择变量  $C_2$ ，其有两个可能的值
- vi 取  $C_2 = 0$
- vii 选择变量  $X_1$ ，其现在只有一个可能的值
- viii 取  $C_1 = 0$
- ix 选择变量  $O$ ，由于  $O + O = R + 10 \times C_1 = R \leq 9$ ，且  $T + T = O + 10 \times F = O + 10$ ，因此  $O$  是小于 5 的偶数，这是最严格的约束
- x 取  $O = 4$
- xi 选择变量  $R$ ，其现在只有一个可能的值

xii 取  $R = 8$

xiii 选择变量  $T$ ，其现在只有一个可能的值

xiv 取  $T = 7$

xv 选择变量  $U$ ，由于  $W + W = U + 10 \times C_2 = U \leq 9$ ，因此  $U$  是小于 9 的偶数

xvi 取  $U = 6$ ，由前向检验限制，不同的字母表示不同的数字，因此  $U$  不能取 4 或 8

xvii 选择变量  $W$ ，其现在只有一个可能的值

xviii 取  $W = 3$

xix 得到一个解： $F = 1, T = 7, U = 6, W = 3, R = 8, O = 4$

## 6 EXERCISE 6.11

用 AC-3 算法说明弧相容对图中问题能够检测出部分赋值  $\{WA = red, V = blue\}$  的不相容

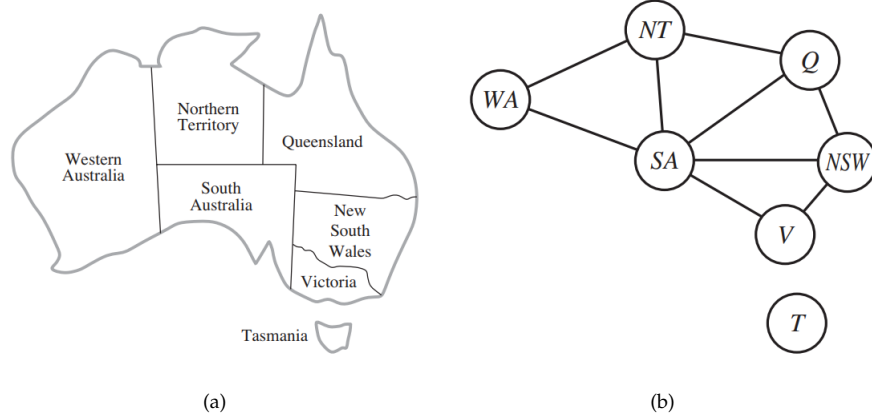


图 5: 着色问题

(a) 澳大利亚的洲和行政区。视此地图着色问题为约束满足问题 (CSP)。目标是对每个区域分配颜色，使得相邻的区域不同色。(b) 将地图着色问题表示成约束图

解 以下列出了 AC-3 算法的执行过程：

- i 删除边  $SA-WA$ ，从  $SA$  的值域中删除  $green$ ，剩余  $\{red, blue\}$
- ii 删除边  $SA-V$ ，从  $SA$  的值域中删除  $red$ ，剩余  $\{blue\}$
- iii 删除边  $NT-WA$ ，从  $NT$  的值域中删除  $green$ ，剩余  $\{red, blue\}$
- iv 删除边  $NT-SA$ ，从  $NT$  的值域中删除  $blue$ ，剩余  $\{red\}$
- v 删除边  $NSW-SA$ ，从  $NSW$  的值域中删除  $blue$ ，剩余  $\{red, green\}$
- vi 删除边  $NSW-V$ ，从  $NSW$  的值域中删除  $green$ ，剩余  $\{green\}$
- vii 删除边  $Q-NT$ ，从  $Q$  的值域中删除  $red$ ，剩余  $\{blue, green\}$
- viii 删除边  $Q-SA$ ，从  $Q$  的值域中删除  $blue$ ，剩余  $\{green\}$
- ix 删除边  $Q-NSW$ ，从  $Q$  的值域中删除  $green$ ，剩余  $\emptyset$ ，返回 **false**

因此部分赋值  $\{WA = red, V = blue\}$  不相容。

## 7 EXERCISE 6.12

用 AC-3 算法求解树结构 CSP 在最坏情况下的复杂度是多少？

解 在树结构 CSP 中，最坏情况下，考虑每一条弧各一次，因此算法的复杂度是  $O(ED)$ ，其中  $E$  是树结构中边的数量， $D$  是最大的域的大小