A* Algorithm - Report

Nicolas Boileau, Simon Stastny

September 20, 2012

1 General A* algorithm

We decided to create own own implementation of A* algorithm for better learning outome. Our general A* algorithm implementation is found in the package `edu.ntnu.simonst.tdt4136.astar` and contains following classes:

- Class `BestFirstSearch` with general A* algorithm implementation
- Class `SearchNode` for general search-node used in the algotithm
- Class `SearchState` for general search-state used in the algotithm
- Class `Fringe` used to store nodes in agenda (list of unexpanded nodes)

Apart from those general classes, this package contains class `App` which is used to run both puzzles from command-line using maven.

To run the puzzles, please use maven command `mvn exec:java` inside the project folder, or run the project within NetBeans (maven enabled).

2 Fractions puzzle

2.1 Initial state of the puzzle

The initial state of the puzzle is state identified by permutation `123456789` which represents following fraction:

$$\frac{1234}{56789}$$

2.2 Description of a goal state

This puzzle involves finding fractions equal to fractions:

$$\frac{1}{2} \quad \frac{1}{3} \quad \frac{1}{4} \quad \frac{1}{5} \quad \frac{1}{6} \quad \frac{1}{7} \quad \frac{1}{8} \quad \frac{1}{9}$$

For example for the first fraction (½) the goal state would have permutation 729314586, because following two fractions are equal:

$$\frac{7293}{14586} = \frac{2^2 \times 3 \times 641}{2^3 \times 3 \times 641} = \frac{1}{2}$$

Similarily for other fractions there exist other permutations of 123456789 which represent fractions equal to them respectively.

2.3 Method of assessing arc costs

Since we are not really interested in the path to the goal node as much as we are interested in the goal-node's state itself, we are not concerned about cost of the solution (lenght of the path). However, we are still trying to work in an optimal way and thus we would like to get to the goal taking as few steps as possible.

For this reason, the arc cost for transition from one state to another is fixed value 1 (in the inner code 100 000 000, because of precision problems related to heuristic function), hence the total cost of path from root to goal is equal to number of steps we have taken.

2.4 Heuristic function description

As long as states are representing fractions, i.e. numbers, the heuristic for estimating cost from current state to goal state could be something so simple as mathematical difference of those two fractions.

Since all the fractions possible in this puzzle have values greater or equal to 0.0124943046625829 and lesser of equal to 0.8, the mathematical difference of any two of them would be a double value somewhere between 0 and 1.

$$\forall x \in possiblefractions, \frac{1234}{98765} \leq x \leq \frac{9876}{12345}$$

We need to map those differences to cardinal numbeers in a way which preserves the order and where no difference would be equal to 0, the smallest difference would be equal to 1 and big differences would be map to big numbers.

Our solution counts the double value of difference of those two fractions, multiplies the value by 100 000 000 and subtracts 10. This results into the smallest difference being assigned estimate of 1, and bigger differences scaled up to somewhere below nearly 80 000 000.

Those big numbers are the reason why the arc costs are multiplied by 1 000 000.

2.5 Successor generation procedure

Successor nodes are generated from current node with a simple switch operator. This switch operator takes the permutation of the node's state (which is a permutation of digits from 1 to 9) and switches two digits to make a new state. Since the first digit can be switched with 8 others to make a new permutations, the second digit with 7 others (because switching with the first digit was done already), third one with 6 others..., this generates 36 (8+7+...+1) new permutations, and each one of them is assigned to a successor node.

This leaves us with 36 successor nodes to each node. At least one of those successor nodes is already closed (because it is current node's parent). We evaluate the remaining nodes' cost estimates and decide which node to expand in the next turn.

2.6 Overview description of a solution

This puzzle is searching for solutions for 8 different fractions, described in 2.2. Those solutions are listed in the subsections below.

2.6.1 Solution for ½

For the first fraction (½) the solution permutation 729314586 was found. This is a solution, because the fraction it represents is equal to goal-state's fraction.

$$\frac{7293}{14586} = \frac{2^2 \times 3 \times 641}{2^3 \times 3 \times 641} = \frac{1}{2}$$

This solution was found in just 5 steps listed here.

- **root** state: 1234/56789 (0.02172956)
- state: 5234/16789 (0.311751742)
- state: 7234/16589 (0.436072096)
- state: 7294/16583 (0.439848037)
- state: 7294/13586 (0.536876196)
- **goal** state: 7293/14586 (0.5)

2.6.2 Solution for ⅓

For the first fraction (⅓) the solution permutation 582317469 was found. This is a solution, because the fraction it represents is equal to goal-state's fraction.

$$\frac{5823}{17469} = \frac{3^2 \times 647}{3^3 \times 647} = \frac{1}{3}$$

.

This solution was found in just 6 steps listed here.

- **root** state: 1234/56789 (0.02172956)
- state: 5234/16789 (0.311751742)

- state: 5234/17689 (0.295890101)
- state: 5834/17629 (0.330931987)
- state: 5834/17269 (0.337830795)
- state: 5824/17369 (0.335310035)
- **goal** state: 5823/17469 (0.333333333)

2.6.3 Solution for ¼

For the first fraction (¼) the solution permutation 579623184 was found. This is a solution, because the fraction it represents is equal to goal-state's fraction.

$$\frac{5796}{23184} = \frac{2^2 \times 3^2 \times 7 \times 23}{2^4 \times 3^2 \times 7 \times 23} = \frac{1}{4}$$

.

This solution was found in just 6 steps listed here.

- **root** state: 1234/56789 (0.02172956)
- state: 1534/26789 (0.057262309)
- state: 7534/26189 (0.287678033)
- state: 5734/26189 (0.218946886)
- state: 5736/24189 (0.237132581)
- state: 5796/24183 (0.239672497)
- **goal** state: 5796/23184 (0.25)

2.6.4 Solution for ⅕

For the first fraction (⅕) the solution permutation 9237/46185 was found. This is a solution, because the fraction it represents is equal to goal-state's fraction.

$$\frac{9237}{46185} = \frac{3 \times 3079}{3 \times 5 \times 3079} = \frac{1}{5}$$

.

This solution was found in just 4 steps listed here.

- **root** state: 1234/56789 (0.02172956)
- state: 9234/56781 (0.162624822)
- state: 9235/46781 (0.197409205)
- state: 9235/46187 (0.199948037)
- **goal** state: 9237/46185 (0.2)

2.6.5 Solution for ⅙

For the first fraction (⅙) the solution permutation 294317658 was found. This is a solution, because the fraction it represents is equal to goal-state's fraction.

$$\frac{2943}{17658} = \frac{3^3 \times 109}{2 \times 3^4 \times 109} = \frac{1}{6}$$

.

This solution was found in just 6 steps listed here.

- **root** state: 1234/56789 (0.02172956)
- state: 2134/56789 (0.0375777)
- state: 2534/16789 (0.150932158)
- state: 2934/16785 (0.174798928)
- state: 2934/17685 (0.165903308)
- state: 2934/17658 (0.166156983)
- **goal** state: 2943/17658 (0.166666667)

2.6.6 Solution for ⅐

For the first fraction (⅐) the solution permutation 761453298 was found. This is a solution, because the fraction it represents is equal to goal-state's fraction.

$$\frac{7614}{53298} = \frac{2 \times 3^4 \times 47}{2 \times 3^4 \times 7 \times 47} = \frac{1}{7}$$

.

This solution was found in just 5 steps listed here.

- **root** state: 1234/56789 (0.02172956)
- state: 7234/56189 (0.12874406)
- state: 7634/52189 (0.146276035)
- state: 7624/53189 (0.143337908)
- state: 7624/53198 (0.143313658)
- **goal** state: 7614/53298 (0.142857143)

2.6.7 Solution for ⅛

For the first fraction (⅛) the solution permutation 817465392 was found. This is a solution, because the fraction it represents is equal to goal-state's fraction.

$$\frac{8174}{65392} = \frac{2 \times 61 \times 67}{2^4 \times 61 \times 67} = \frac{1}{8}$$

.

This solution was found in just 5 steps listed here.

- **root** state: 1234/56789 (0.02172956)
- state: 8234/56719 (0.145171812)
- state: 8234/65719 (0.125291012)
- state: 8134/65729 (0.123750552)
- state: 8134/65792 (0.123632053)
- **goal** state: 8174/65392 (0.125)

2.6.8 Solution for ⅑

For the first fraction (⅑) the solution permutation 638157429 was found. This is a solution, because the fraction it represents is equal to goal-state's fraction.

$$\frac{6381}{57429} = \frac{3^2 \times 709}{3^4 \times 709} = \frac{1}{9}$$

.

This solution was found in just 5 steps listed here.

- **root** state: 1234/56789 (0.02172956)
- state: 6234/51789 (0.120373052)
- state: 6234/57189 (0.109006977)
- state: 6231/57489 (0.108385952)
- state: 6831/57429 (0.118946874)
- **goal** state: 6381/57429 (0.111111111)

3 Checkers puzzle

1. The initial state of the puzzle. initial state: BBB AAA, respective BBBBBB AAAAAA, BBBBBBBBBBBB AAAAAAAAAAAA

2. A general description of a goal state. goal state: AAA BBB, respective AAAAAA BBBBBB, AAAAAAAAAAAA BBBBBBBBBBBB

3. The method of assessing arc costs. It may be a

xed value or a more complex procedure. arc costs - fixed value 1

4. A clear, concise description (using mathematical expressions and text) of the heuristic function (h) used to solve the puzzle. heuristic function see CheckersState.getDistance(a,b)

5. A thorough description of the procedure used to generate successor states when expanding a node. see CheckersNode.getChildren()

6. An overview description of a solution (i.e. path from start to goal) found by A*. The sequence of search states from the start to the goal node must be presented along with a brief summary (in Norwegian or English) of the main state-to-state transitions within sequence.

see output from program..