CS2310 Term Project Report

Angen Zheng
angen.zheng@gmail.com

Title

8-Puzzle Problem in SIS Testbed

Description

Problem: 8-Puzzle Problem is a game played on a 3-by-3 grid with 8 square blocks labeled 1 through 8 and a blank square (See Fig. 1). Your goal is to rearrange the blocks so that they are in order running around the perimeter of the board. You are permitted to slide blocks horizontally or vertically into the blank square. The following figure shows a sequence of legal moves from an initial configuration (left) to the goal configuration (right).

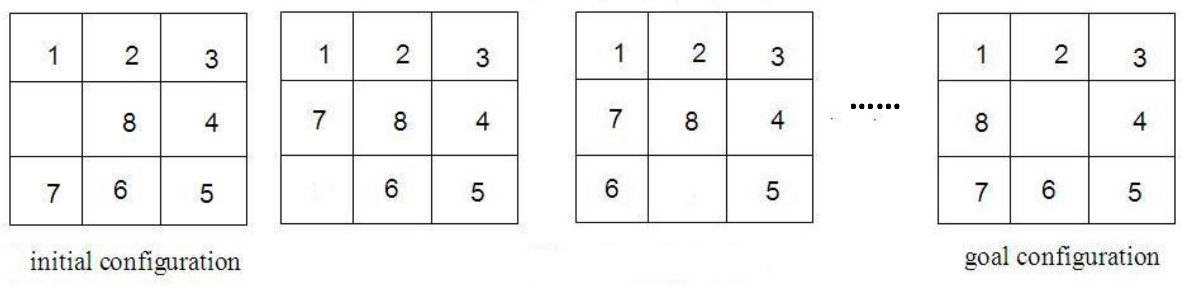


Fig. 1

Project Overview:

- First, write a NextMove(X) program, which accepts input problem set, to produce an output set Y within one step.
- Second, run the above program in the Testbed with following cycle switching rule:

  cycle0: P0 -enum< P1 < -enum < P2 < -enum P3….. < -enum < Pn-1 >elim- Pn

  When the final eliminate operator an empty output set Pn, then cycle0 is reran with Pn-1 as its initial problem set; otherwise cycle0 is terminated. In our implementation, the user can specify the initial configuration and the number of enumerate operators the Data\data1.txt and System\Config\Config.txt file.

- Third, enhance the Testbed, making it automatically support the above cycle switching rule. Previously, we need to modify DataSender, Verifier, and Eliminator three system components as well as the application component generated by SISProjectCreator, now we only need to update two the environment variables in the application component code generated by SISProjectCreator to specify whether you want to rerun current cycle with a new input.

How to Run the SIS 8-Puzzle Project

a. Use SIS Version v3n5

b. Run Scripts\runServer.bat to start SISServer

c. Run prjRemote.jar to start prjRemote

    1) Press the Connect button

    2) Load xml files in 8puzzle\sis8puzzle\xml\InitXML\ and send them one by one
    Or

    3) Load list_InitXML.txt in 8puzzle\sis8puzzle\xml\InitXML\ and send All

d. If you are using Windows operating system, you can run:

    1) runDataSender.bat

    2) runEliminator.bat

    3) runEnumerator.bat

    4) runVerifier.bat

    5) runTimeController.bat

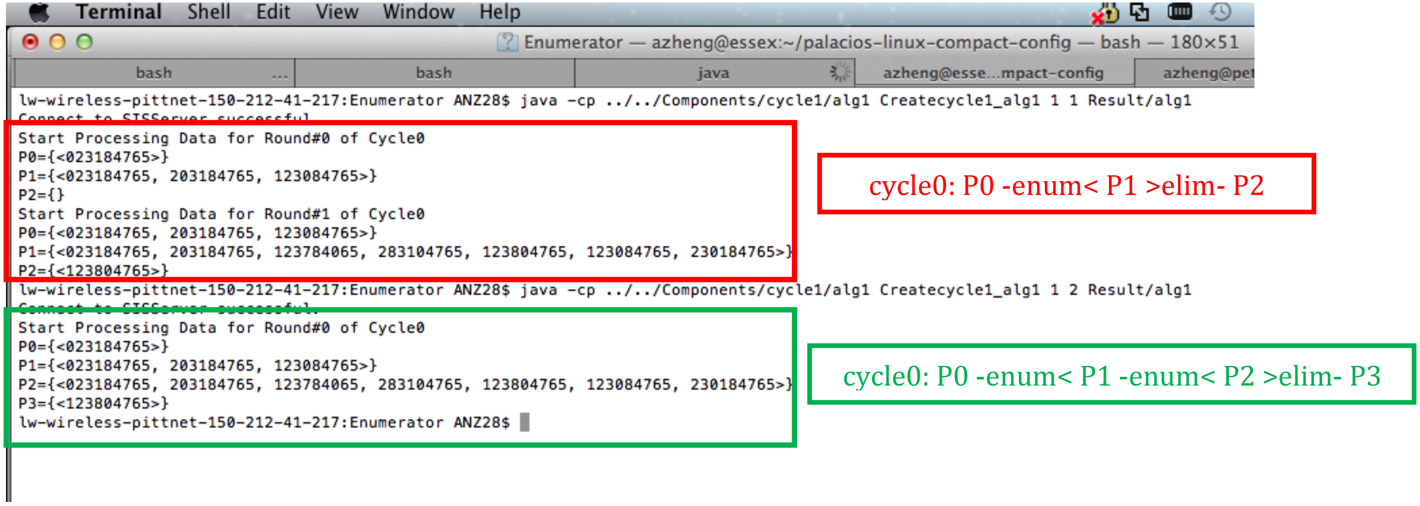
    to start all system components. If you are running on linux or mac operating systems, you can also run 8puzzle\sis8puzzle\System\runSysComp.sh to start all system components.

e. You can modify the initial configuration state in file 8puzzle\sis8puzzle\data\data1.txt.

f. You can check the full trace in file 8puzzle\sis8puzzle\data\fulltrace.txt to see all the problems genterated from initial problem set to goal problem set.

g. You can modify the number of enumerate operators in file 8puzzle\sis8puzzle\System\Config\Config.txt in the following section:

```
BEGIN CycleAlgorithm cycle1
/*This section lists cycle algorithm information.*/
/*The format is AlgorithmName \t AlgorithmPath \t number of enumerate operators.*/
alg1      cycle1/alg1/Createcycle1_alg1   1
END CycleAlgorithm
```


Terminal
Shell
Edit
View
Window
Help
Enumerator — azheng@essex:~/palacios-linux-compact-config — bash — 180×51
bash
...
bash
java
azheng@esse...mpact-config
azheng@pe

```
lw-wireless-pittnet-150-212-41-217:Enumerator ANZ28$ java -cp ../../Components/cycle1/alg1 Createcycle1_alg1 1 1 Result/alg1
Connect to SISServer successful
Start Processing Data for Round#0 of Cycle0
P0={<023184765>}
P1={<023184765, 203184765, 123084765>}
P2={}
Start Processing Data for Round#1 of Cycle0
P0={<023184765, 203184765, 123084765>}
P1={<023184765, 203184765, 123784065, 283104765, 123804765, 123084765, 230184765>}
P2={<123804765>}
lw-wireless-pittnet-150-212-41-217:Enumerator ANZ28$ java -cp ../../Components/cycle1/alg1 Createcycle1_alg1 1 2 Result/alg1
Connect to SISServer successful.
Start Processing Data for Round#0 of Cycle0
P0={<023184765>}
P1={<023184765, 203184765, 123084765>}
P2={<023184765, 203184765, 123784065, 283104765, 123804765, 123084765, 230184765>}
P3={<123804765>}
lw-wireless-pittnet-150-212-41-217:Enumerator ANZ28$
```

cycle0: P0 -enum< P1 >elim- P2

cycle0: P0 -enum< P1 -enum< P2 >elim- P3

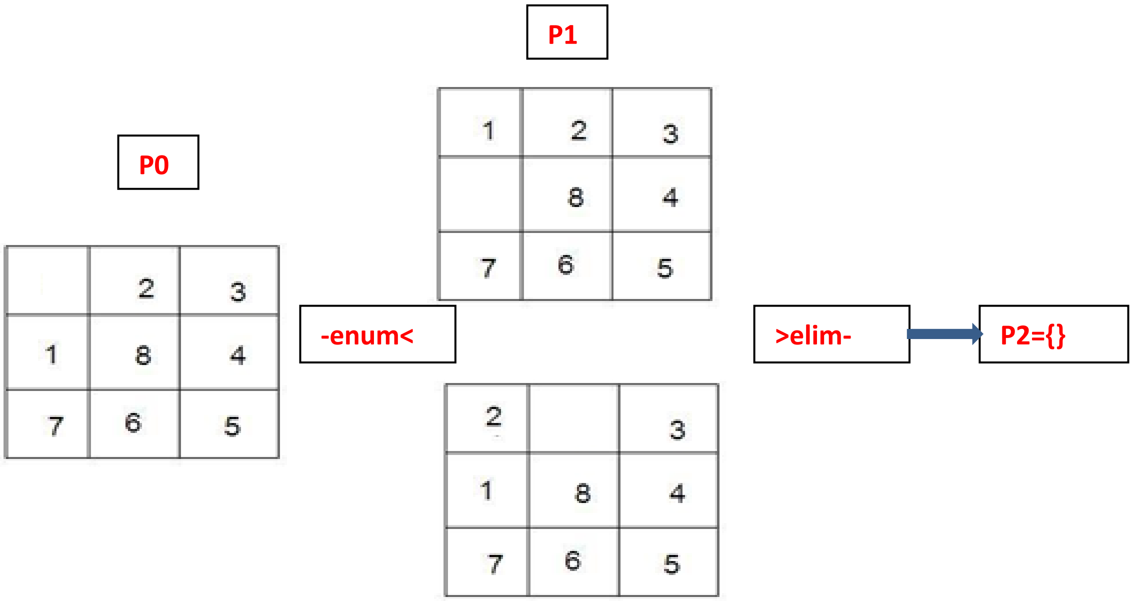
(a screenshot of the experiment results)

The above figure shows the results of running 8-puzzle with following two cycles with 023184765 as their initial problem set respectively:

cycle0: P0 -enum< P1 >elim- P2

cycle0: P0 -enum< P1 -enum< P2 >elim- P3

As we can see from the figure, the red cycle requires two rounds since it only has one enumerate operator, while the green cycle only needs one round thanks to its two enumerate operators. The following figure shows the first round execution of cycle0 (Red Cycle). At the end of this round, the cycle will be restart with P1 as its initial problem set.



How to Create Single-Cycle SIS Projects with following Cycle Switching Rule.

That is, to create an SIS project implementing following cycle switching rule:

Cycle0: P0 -enum< P1 -enum< P2... -enum< Pn-1 >elim- Pn

When the final eliminate operator an empty output set Pn, then cycle0 is reran with Pn-1 as its initial problem set; else cycle0 is terminated.

a. Replace the template code ds, ver, el, eu and siscomp in Testbed\SISProjectCreator\Templates\ with the new code in 8puzzle\New Template Codes\.
b. Run SISProjectCreator:

```
cd SISProjectCreator
java -jar SISProjectCreator.jar
```

c. Create a single-cycle SIS project as usual (see Testbed\sisv3n5parallel\sisv3n5parallel.doc).

d. Made necessary changes to Component generated by SISProjectCreator, update rerunCurCycle and rerunInput variables in ProcessMsg method related to 601 message processing based on your processing results. The rerunCurCycle indicates whether you want to rerun current cycle, while the rerunInput variable specifies the input for the rerun. In default, rerunCurCycle is set to false.

e. Run the SIS app as usual.