

PATIA Report

Mingming DAI

May 2022

1 Documentation for code and experiment execution

Compile the code with the following command under pddl4j folder :

```
javac -d classes -cp classes:lib/pddl4j-4.0.0.jar  
↪ src/main/java/fr/uga/pddl4j/examples/*.java
```

To run a specific problem using asp, use the following command under pddl4j folder :

```
java -cp classes:lib/pddl4j-4.0.0.jar  
↪ fr.uga.pddl4j.examples.ASP domainPath problemPath -e  
↪ FAST\_FORWARD -w 1.2 -t 300
```

To run a specific problem using arvand static, use the following command under pddl4j folder :

```
java -cp classes:lib/pddl4j-4.0.0.jar  
↪ fr.uga.pddl4j.examples.Arvand_static domainPath  
↪ problemPath -e FAST\_FORWARD -w 1.2 -t 300
```

To run a specific problem using arvand dynamic, use the following command under pddl4j folder :

```
java -cp classes:lib/pddl4j-4.0.0.jar  
↪ fr.uga.pddl4j.examples.Arvand_dynamic domainPath  
↪ problemPath -e FAST\_FORWARD -w 1.2 -t 300
```

For testing purpose, I also provided a script file runBenchmarks.sh which can execute and validate multiple problems in the same domain.

```
bash runbenchmark.sh arg1 arg2  
arg1 can be one of the six following : gripper, logistics,  
↪ mprime, mystery, ipc3, ipc2002depots  
arg2 can be one of the three following : static, dynamic, asp
```

To plot the results respectively for metrics "Resolution time" and "makeplan":

```
+ Expe python plotResults.py
usage: plotResults.py [-h] [-d DOMAINE] [-m MODE]

options:
  -h, --help            show this help message and exit
  -d DOMAINE, --domaine DOMAINE
                        Full domaine name (default: gripper)
  -m MODE, --mode MODE  Choose to plot which metrics, r for resolution time or
                        m for makespan (default: r)
```

Figure 1: usage of the .py file

2 Remarks on experiments

2.1 Gripper

For testing experiments I first run the test on gripper with :

```
bash runbenchmark.sh gripper asp
bash runbenchmark.sh gripper static
bash runbenchmark.sh gripper dynamic
```

I end up with the following statistics with commands:

```
python plotResults.py -d gripper -m p
python plotResults.py -d gripper -m r
```

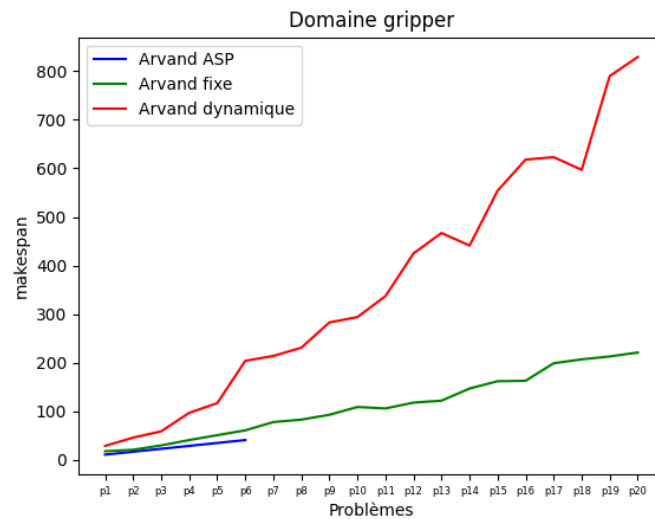


Figure 2: gripper makespan

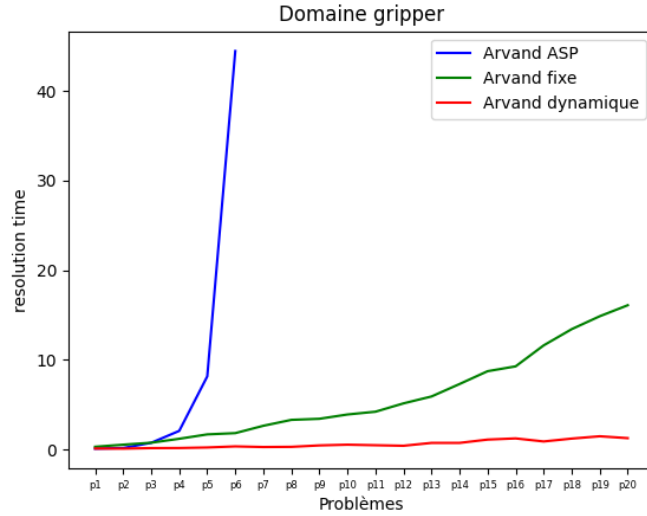


Figure 3: gripper resolution time

Note that all the problem after P06 with ASP took too long to run (more than 20 mins with timeout = 300 using an Intel i5-8600K CPU with 6 cores, 12 thread and a clock speed of 4.3 GHz), and it returns no plan.

2.2 Logistics

Then run the test on logistics with :

```
bash runbenchmark.sh logistics asp
bash runbenchmark.sh logistics static
bash runbenchmark.sh logistics dynamic
```

Here's the result obtained and plotted:

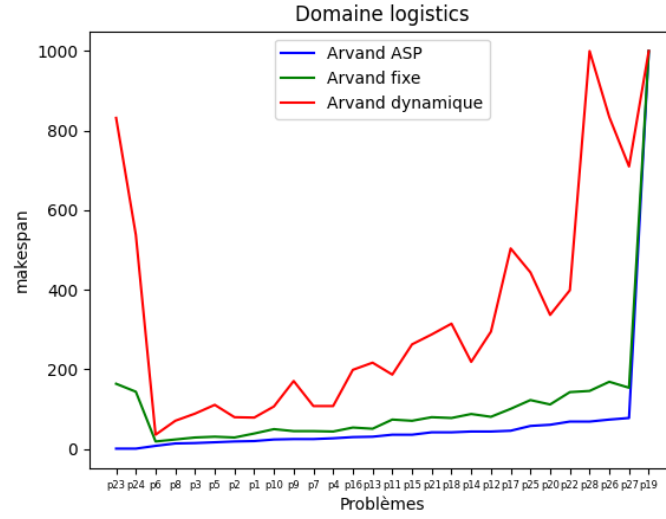


Figure 4: logistics makespan

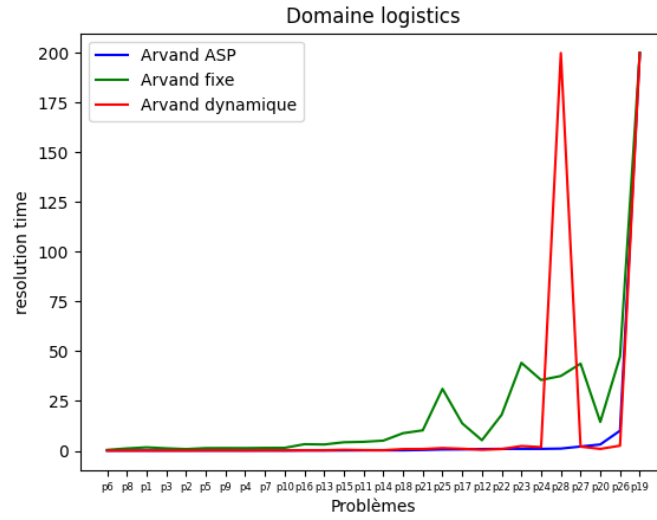


Figure 5: logistics resolution time

While testing logistics, I noticed that some tests take very long time to solve, so I manually give them a makespan of 1000 and a resolution time of 200 seconds.

2.3 mprime

For mprime I did try to run the test, but some problems take too long to run. And some problems couldn't find acceptable solutions before timeout. Due to the considerable missing of valid results, I chose not to plot them. Instead, here's the output of ASP on mprime:

```
Test 1 :
Final value: 5
0.23 seconds total time
2.66 MBytes total

Test 2 :
Final value: 8
0.48 seconds total time
9.24 MBytes total

Test 3 :
Final value: 4
0.37 seconds total time
7.08 MBytes total

Test 4 :
Final value: 9
0.25 seconds total time
3.15 MBytes total

Test 5 :
Final value: 11
3.38 seconds total time
10.17 MBytes total

Test 6 :
Final value: 13
198.82 seconds total time
66.43 MBytes total

Test 7 :
Final value: 5
0.32 seconds total time
3.93 MBytes total

Test 8 :
Final value: 6
1.70 seconds total time
22.34 MBytes total

Test 9 :
Final value: 8
0.47 seconds total time
6.44 MBytes total

Test 10 :
1.92 seconds total time
117.15 MBytes total

Test 11 :
```

```

Final value: 7
          0.27 seconds total time
          3.49 MBytes total
Test 12 :
Final value: 7
          1.04 seconds total time
          6.17 MBytes total
Test 13 :
          1.59 seconds total time
          110.89 MBytes total
Test 14 :
          1.91 seconds total time
          149.03 MBytes total
Test 15 :
          1.40 seconds total time
          102.56 MBytes total
Test 16 :
Final value: 6
          0.99 seconds total time
          16.00 MBytes total
Test 17 :
Final value: 4
          1.78 seconds total time
          54.52 MBytes total
Test 18 :
          1.10 seconds total time
          69.32 MBytes total
Test 19 :
Final value: 7
          66.14 seconds total time
          50.25 MBytes total
Test 20 :
Test 21 :
Test 22 :
Test 23 :
Test 24 :
Test 25 :
Test 26 :
Test 27 :
Test 28 :
Test 29 :
Test 30 :

```

And here's the output of arvand static:

```

Test 1 :
Final value: 17

```

	2.51 seconds total time
	2.65 MBytes total
Test 2 :	
Final value: 22	
	24.61 seconds total time
	9.16 MBytes total
Test 3 :	
Final value: 18	
	9.30 seconds total time
	7.04 MBytes total
Test 4 :	
Final value: 29	
	6.76 seconds total time
	3.09 MBytes total
Test 5 :	
Final value: 40	
	21.81 seconds total time
	7.47 MBytes total
Test 6 :	
Final value: 77	
	901.90 seconds total time
	56.19 MBytes total
Test 7 :	
Final value: 9	
	2.46 seconds total time
	3.90 MBytes total
Test 8 :	
Final value: 30	
	92.15 seconds total time
	22.16 MBytes total
Test 9 :	
Final value: 22	
	13.75 seconds total time
	6.41 MBytes total
Test 10 :	
Test 11 :	
Test 12 :	
Test 13 :	
Test 14 :	
Test 15 :	
Test 16 :	
Test 17 :	
Test 18 :	
Test 19 :	
Test 20 :	
Test 21 :	

Test 22 :
 Test 23 :
 Test 24 :
 Test 25 :
 Test 26 :
 Test 27 :
 Test 28 :
 Test 29 :
 Test 30 :

2.4 ipc2002-depot

From the fourth problem of this domain, it took longer and longer to return the results. I didn't have enough time waiting for all the problems to be done.

2.5 Sokoban

Both arvand dynamic and static took way too long to run (more than 15 minutes) and still unable to find an acceptable plan, so I stopped them manually. ASP did return a plan within a reasonable amount of time. Here's the plan obtained by using ASP:

* Starting A* search
 * A* search succeeded

found plan as follows:

```
000: (      move pos-6-4 pos-7-4 d) [0]
001: (      move pos-7-4 pos-7-3 l) [0]
002: (      move pos-7-3 pos-7-2 l) [0]
003: (      move pos-7-2 pos-6-2 u) [0]
004: (      move pos-6-2 pos-5-2 u) [0]
005: (      move pos-5-2 pos-5-1 l) [0]
006: (      move pos-5-1 pos-4-1 u) [0]
007: (push pos-4-1 pos-4-2 pos-4-3 r) [0]
008: (      move pos-4-2 pos-5-2 d) [0]
009: (      move pos-5-2 pos-6-2 d) [0]
010: (      move pos-6-2 pos-7-2 d) [0]
011: (      move pos-7-2 pos-7-3 r) [0]
012: (      move pos-7-3 pos-7-4 r) [0]
013: (      move pos-7-4 pos-6-4 u) [0]
014: (      move pos-6-4 pos-5-4 u) [0]
015: (push pos-5-4 pos-5-3 pos-5-2 l) [0]
016: (      move pos-5-3 pos-5-4 r) [0]
017: (push pos-5-4 pos-4-4 pos-3-4 u) [0]
018: (      move pos-4-4 pos-4-5 r) [0]
```



```

019: (      move pos-4-5 pos-4-6 r) [0]
020: (      move pos-4-6 pos-5-6 d) [0]
021: (push pos-5-6 pos-5-5 pos-5-4 l) [0]
022: (push pos-5-5 pos-5-4 pos-5-3 l) [0]
023: (      move pos-5-4 pos-4-4 u) [0]
024: (push pos-4-4 pos-3-4 pos-2-4 u) [0]
025: (      move pos-3-4 pos-3-3 l) [0]
026: (      move pos-3-3 pos-3-2 l) [0]
027: (      move pos-3-2 pos-4-2 d) [0]
028: (push pos-4-2 pos-4-3 pos-4-4 r) [0]
029: (push pos-4-3 pos-4-4 pos-4-5 r) [0]
030: (      move pos-4-4 pos-5-4 d) [0]
031: (      move pos-5-4 pos-6-4 d) [0]
032: (      move pos-6-4 pos-7-4 d) [0]
033: (      move pos-7-4 pos-7-3 l) [0]
034: (      move pos-7-3 pos-7-2 l) [0]
035: (      move pos-7-2 pos-6-2 u) [0]
036: (push pos-6-2 pos-5-2 pos-4-2 u) [0]
037: (      move pos-5-2 pos-5-1 l) [0]
038: (      move pos-5-1 pos-4-1 u) [0]
039: (push pos-4-1 pos-4-2 pos-4-3 r) [0]
040: (      move pos-4-2 pos-3-2 u) [0]
041: (      move pos-3-2 pos-3-3 r) [0]
042: (      move pos-3-3 pos-3-4 r) [0]
043: (      move pos-3-4 pos-4-4 d) [0]
044: (      move pos-4-4 pos-5-4 d) [0]
045: (push pos-5-4 pos-5-3 pos-5-2 l) [0]
046: (      move pos-5-3 pos-5-4 r) [0]
047: (      move pos-5-4 pos-6-4 d) [0]
048: (      move pos-6-4 pos-7-4 d) [0]
049: (      move pos-7-4 pos-7-3 l) [0]
050: (      move pos-7-3 pos-7-2 l) [0]
051: (      move pos-7-2 pos-6-2 u) [0]
052: (push pos-6-2 pos-5-2 pos-4-2 u) [0]
053: (      move pos-5-2 pos-5-3 r) [0]
054: (      move pos-5-3 pos-5-4 r) [0]
055: (      move pos-5-4 pos-4-4 u) [0]
056: (      move pos-4-4 pos-3-4 u) [0]
057: (      move pos-3-4 pos-3-3 l) [0]
058: (      move pos-3-3 pos-2-3 u) [0]
059: (      move pos-2-3 pos-1-3 u) [0]
060: (      move pos-1-3 pos-1-4 r) [0]
061: (push pos-1-4 pos-2-4 pos-3-4 d) [0]
062: (push pos-2-4 pos-3-4 pos-4-4 d) [0]
063: (push pos-3-4 pos-4-4 pos-5-4 d) [0]
064: (push pos-4-4 pos-5-4 pos-6-4 d) [0]

```

```

065: (      move pos-5-4 pos-5-3 l) [0]
066: (push pos-5-3 pos-4-3 pos-3-3 u) [0]
067: (push pos-4-3 pos-3-3 pos-2-3 u) [0]
068: (      move pos-3-3 pos-4-3 d) [0]
069: (      move pos-4-3 pos-5-3 d) [0]
070: (      move pos-5-3 pos-5-4 r) [0]
071: (      move pos-5-4 pos-5-5 r) [0]
072: (      move pos-5-5 pos-5-6 r) [0]
073: (      move pos-5-6 pos-4-6 u) [0]
074: (push pos-4-6 pos-4-5 pos-4-4 l) [0]
075: (push pos-4-5 pos-4-4 pos-4-3 l) [0]
076: (      move pos-4-4 pos-5-4 d) [0]
077: (      move pos-5-4 pos-5-3 l) [0]
078: (      move pos-5-3 pos-5-2 l) [0]
079: (      move pos-5-2 pos-6-2 d) [0]
080: (      move pos-6-2 pos-7-2 d) [0]
081: (      move pos-7-2 pos-7-3 r) [0]
082: (      move pos-7-3 pos-7-4 r) [0]
083: (push pos-7-4 pos-6-4 pos-5-4 u) [0]
084: (      move pos-6-4 pos-7-4 d) [0]
085: (      move pos-7-4 pos-7-3 l) [0]
086: (      move pos-7-3 pos-7-2 l) [0]
087: (      move pos-7-2 pos-6-2 u) [0]
088: (      move pos-6-2 pos-5-2 u) [0]
089: (      move pos-5-2 pos-5-3 r) [0]
090: (push pos-5-3 pos-5-4 pos-5-5 r) [0]
091: (      move pos-5-4 pos-4-4 u) [0]
092: (      move pos-4-4 pos-4-5 r) [0]
093: (      move pos-4-5 pos-4-6 r) [0]
094: (      move pos-4-6 pos-5-6 d) [0]
095: (push pos-5-6 pos-5-5 pos-5-4 l) [0]
096: (push pos-5-5 pos-5-4 pos-5-3 l) [0]
097: (      move pos-5-4 pos-4-4 u) [0]
098: (      move pos-4-4 pos-3-4 u) [0]
099: (      move pos-3-4 pos-3-3 l) [0]
100: (      move pos-3-3 pos-3-2 l) [0]
101: (push pos-3-2 pos-4-2 pos-5-2 d) [0]
102: (push pos-4-2 pos-4-3 pos-4-4 r) [0]
103: (push pos-4-3 pos-4-4 pos-4-5 r) [0]
104: (      move pos-4-4 pos-3-4 u) [0]
105: (      move pos-3-4 pos-2-4 u) [0]
106: (      move pos-2-4 pos-1-4 u) [0]
107: (      move pos-1-4 pos-1-3 l) [0]
108: (push pos-1-3 pos-2-3 pos-3-3 d) [0]
109: (      move pos-2-3 pos-2-4 r) [0]
110: (      move pos-2-4 pos-3-4 d) [0]

```

```
111: (      move pos-3-4 pos-4-4 d) [0]
112: (      move pos-4-4 pos-5-4 d) [0]
113: (      move pos-5-4 pos-6-4 d) [0]
114: (      move pos-6-4 pos-7-4 d) [0]
115: (      move pos-7-4 pos-7-3 l) [0]
116: (      move pos-7-3 pos-7-2 l) [0]
117: (      move pos-7-2 pos-6-2 u) [0]
118: (push pos-6-2 pos-5-2 pos-4-2 u) [0]
119: (      move pos-5-2 pos-5-1 l) [0]
120: (      move pos-5-1 pos-4-1 u) [0]
121: (push pos-4-1 pos-4-2 pos-4-3 r) [0]
```

```
time spent:      0.05 seconds parsing
                 0.40 seconds encoding
                 7.32 seconds searching
                 7.77 seconds total time
```

```
memory used:     2.29 MBytes for problem representation
                 25.36 MBytes for searching
                 27.65 MBytes total
```