On the Evolution of Planner-Specific Macro Sets

Mauro Vallati, Lukáš Chrpa, Ivan Serina
University of Huddersfield,Czech Technical University in Prague,
Charles University in Prague, Univeristà degli Studi di Brescia

Introduction

- Macros are a well-established reformulation technique for encapsulating sequences of operators in AI-planning
- Existing techniques require a training phase, and rely on the fact that training is representative of testing instances
- Extracted macros can not be modified and are used on every instance of the considered domain

Idea

We propose an approach capable of:

- Automatically combine macros, collected from different sources, in suitable sets
- Evolve the composition of macro sets according to observed performance of a given planner
- Exploit the availability of multi-core machines

The MEvo Approach

The core of MEvo is presented in Algorithm 1, and requires a domain model **D**, a stream of problems **I**, a solver **p**, a pool of macros **M**, and a the max number of macros per set **n**.

It operates by solving each problem via 4 set of macros:

- **Original** (O): empty set, for solvability.
- **Random**: a randomly selected set of macros.
- **Best** (B): the best macros, according to observed performance.
- **AlmostBest** (A): a mix of best and near-best macros.

Algorithm 1 Components of the MEvo algorithm

```
1: procedure MEvo(M,I,D,p,n)
2:     E = initialiseMacroScores(M)
3:     for i in I do
4:         S = getMacroSets(E,M,n)
5:         r = solveInstance(S,D,i,p)
6:         E = updateMacroScores(r,E)
7:     end for
8: end procedure
```

Macros' score is updated after each iteration, and takes into account the number of macros in each set, and the performance achieved.

Results and Take-home Message

- MEvo allows to effectively select and exploit macros for improving the performance of planners.
- Planners tend to prefer different macros, according to the structure of the problems and their search strategy
- New macros can be added overtime, without restarting.
- Macro sets can be efficiently evolved, also in case of sudden structural problems changes.
- MEvo provides a good approach for exploiting multi-core machines.
- MEvo outperforms state of the art of multicore planning.

Set		Domains				
		Bar 150	Bw 160	Dep 165	MBW 150	Park 140
Jasper	O	94.3 (69.3)	57.2 (36.9)	84.6 (55.2)	**36.8** (26.7)	0.0 (0.0)
	B	113.6 (80.7)	**86.2** (55.0)	**86.1** (56.3)	34.7 (26.0)	0.0 (0.0)
	A	**115.1** (80.7)	40.3 (31.3)	78.9 (52.1)	34.3 (26.)	0.0 (0.0)
	MEvo	130.0 (86.7)	97.0 (60.6)	120.0 (72.7)	49.0 (32.7)	0.0 (0.0)
LPG	O	0.0 (0.0)	70.2 (76.9)	**146.2** (100.0)	**24.2** (17.3)	0.0 (0.0)
	B	0.0 (0.0)	**158.4** (100.0)	107.6 (95.2)	22.5 (18.7)	0.0 (0.0)
	A	0.0 (0.0)	98.7 (63.1)	111.6 (100.0)	23.2 (19.3)	0.0 (0.0)
	MEvo	0.0 (0.0)	160.0 (100.0)	165.0 (100.0)	33.0 (22.0)	0.0 (0.0)
Mp	O	13.5 (10.7)	0.0 (0.0)	147.8 (100.0)	**1.2** (1.3)	1.0 (0.7)
	B	**15.5** (11.3)	**98.5** (62.5)	**162.8** (100.0)	1.1 (1.3)	1.0 (0.7)
	A	14.5 (10.7)	47.8 (31.3)	159.9 (98.8)	1.0 (0.7)	0.0 (0.0)
	MEvo	24.0 (16.0)	106.0 (66.3)	165.0 (100.0)	3.0 (2.0)	1.0 (0.7)
Probe	O	89.6 (68.0)	106.6 (77.5)	**164.5** (100.0)	37.8 (27.3)	2.0 (1.4)
	B	**98.4** (73.3)	**140.8** (88.8)	159.0 (100.0)	**42.2** (32.7)	2.0 (1.4)
	A	93.5 (71.3)	135.2 (86.3)	157.6 (100.0)	34.7 (27.3)	0.0 (0.0)
	MEvo	128.0 (85.3)	144.0 (90.0)	165.0 (100.0)	64.0 (42.7)	2.0 (1.4)
Yahsp	O	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	6.9 (4.7)	0.0 (0.0)
	B	0.0 (0.0)	0.0 (0.0)	3.7 (2.4)	7.0 (4.7)	0.0 (0.0)
	A	0.0 (0.0)	0.0 (0.0)	**8.0** (4.8)	7.0 (4.7)	0.0 (0.0)
	MEvo	0.0 (0.0)	0.0 (0.0)	11.0 (6.7)	7.0 (4.7)	0.0 (0.0)