Efficient Black-Box Planning Using Macro-Actions with Focused Effects

Cameron Allen[1,2], Michael Katz[2], Tim Klinger[2], George Konidaris[1], Matthew Riemer[2], Gerald Tesauro[2]

[1] Brown University [2] IBM Research

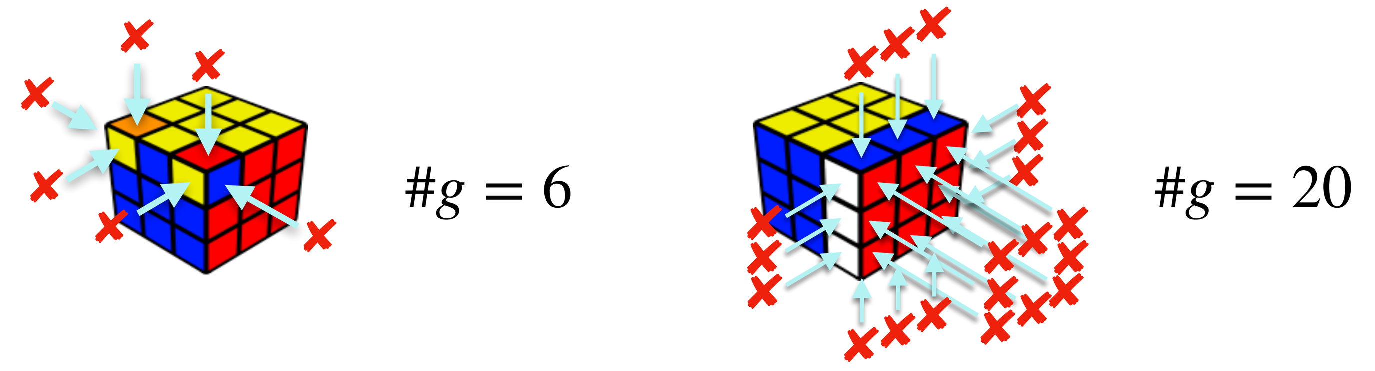
csal@brown.edu
@camall3n

Black-Box Planning

Black-box planning agents don't have declarative action descriptions. Instead, they must query a simulator to see which actions $A(s)$ are applicable in a given state, and what are their effects, $\mathrm{Sim}(s, a)$. Each state s is a vector of finite-valued state variables, and the goal G is a list of variable assignments to all (or a subset) of the state variables. This means goal-aware heuristics are often limited to goal-counting.

The Goal-Count Heuristic

The goal-count heuristic, $\#g$, is a simple (often uninformative) domain-independent heuristic that is compatible with black-box planning, which counts how many state variables differ from their goal values.

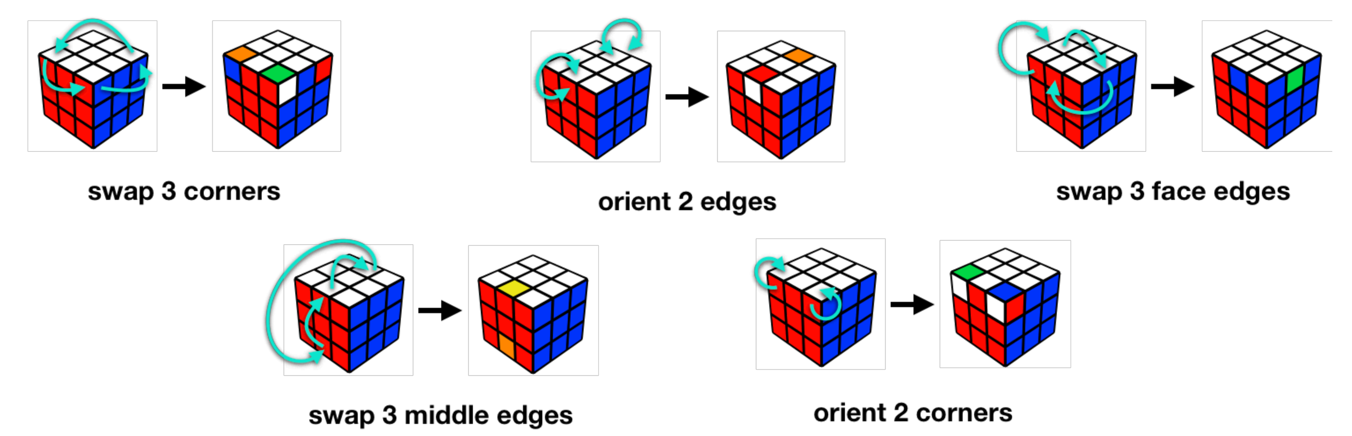


Macro-Actions and Effect Size

A ***macro-action*** (or ***macro***) is a deterministic sequence of actions, typically for the purpose of accomplishing some useful subgoal. The ***effect size*** of a macro is the number of state variables, measured at the end of macro execution, that are different from their starting values. We call macros that only modify a small number of state variables ***focused***.

Expert Macros Are Focused & Make Goal Counting More Accurate

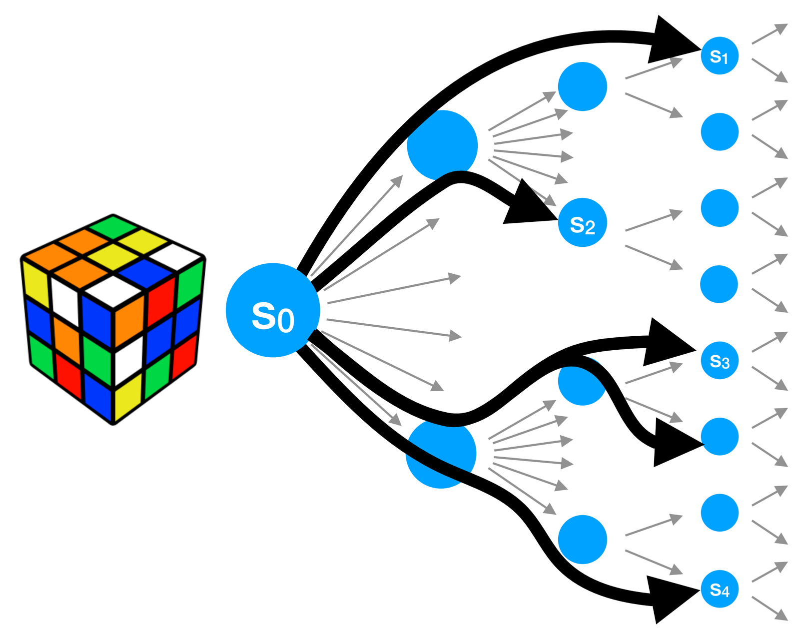
Expert macros modify specific state variables, and minimize side-effects to others. This makes progress towards the goal more predictable as macros are less likely to "undo" previous work.



Adding ***focused*** macro-actions, that **modify few state variables**, makes black-box planning **more efficient**.

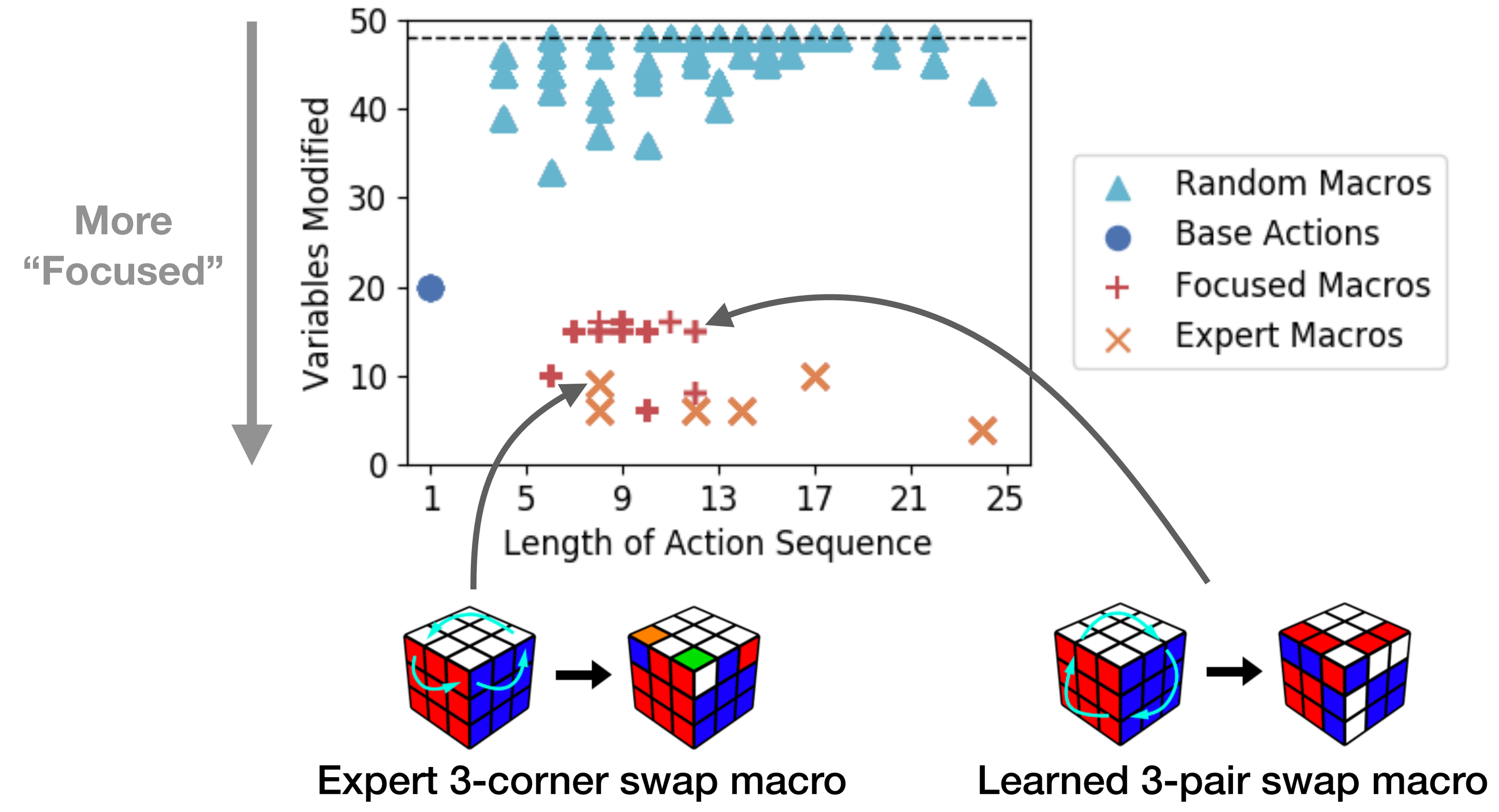
Discovering Focused Macro-Actions

From an arbitrary initial state s_0, we can learn focused macros as follows:

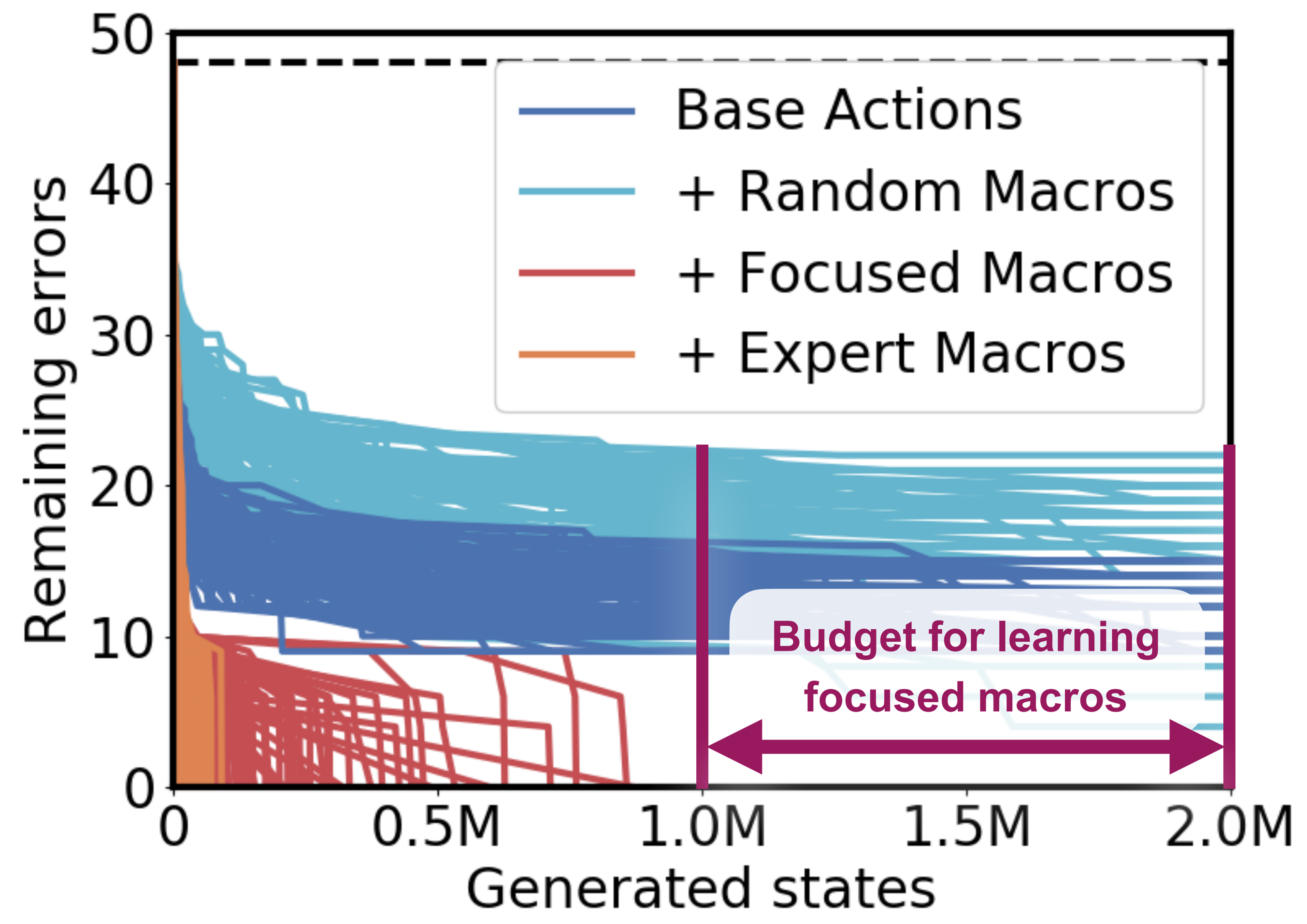


- For each candidate macro, measure its effect size:
  - $s_0 \rightarrow s_1$: 14 variables
  - $\boxed{s_0 \rightarrow s_2 : \text{8 variables}}$ Focused!
  - $s_0 \rightarrow s_3$: 16 variables
  - etc…
- Best-first search (expand node w/ lowest effect size)
- Store the N best macros

Focused Macros Are Like Expert Macros



Focused Macros Speed Up Planning



Focused Macros Help for Many Domains

Domain	N_M	B_M	GBFS(A) Gen	GBFS(A) Sol	GBFS(A+M) Gen	GBFS(A+M) Sol	BFWS(A) Gen	BFWS(A) Sol	BFWS(A+M) Gen	BFWS(A+M) Sol	LAMA(A) Gen	LAMA(A) Sol
Depot	8	50K	58275.9	**0.74**	**55132.4**	0.60	75966.9	**0.48**	**72205.8**	0.34	46620.9	1.00
Doors	8	5K	3050.7	1.00	**512.6**	1.00	4660.9	1.00	**3057.3**	1.00	293.0	1.00
Ferry	8	5K	1875.8	1.00	**1151.4**	1.00	1209.9	1.00	**1163.5**	1.00	699.8	1.00
Gripper	8	5K	7314.8	1.00	**6277.0**	1.00	44945.9	1.00	**6295.9**	1.00	6493.1	1.00
Hanoi	8	100K	78433.6	0.78	**6358.8**	**1.00**	63455.2	1.00	**3365.9**	1.00	65496.4	1.00
Miconic	8	5K	7559.4	1.00	**1907.1**	1.00	10269.2	1.00	**1884.3**	1.00	1316.7	1.00
15-Puz.	192	32K	30840.5	1.00	**4952.4**	1.00	109425.2	1.00	**6290.1**	1.00	–	–
Rubik's	576	1M	>2M	0.00	**171.3K**	**1.00**	>2M	0.00	**163.8K**	**1.00**	9.13M	1.00

Table 2: Black-box planning results for PDDLGym-based simulators (top), and domain-specific simulators (bottom). (A) - primitive actions only; (A+M) - primitive actions + focused macros; N_M - number of macros; B_M - macro-learning budget; Gen - generated states; Sol - solve rate; (bold) - best performance of each planner. The efficiency of both GBFS and BFWS(R_G^*) are improved by adding focused macros. Note that LAMA is an informed planner with access to much more information than black-box planners, and is only included for reference.