

The solution of a CSP requires that a value is found for each of the variables, and that such values are taken from the respective domains of those variables. Additionally, the

set of values assigned must simultaneously satisfy all of the declared constraints. However, certain values for a given variable may be *interchangeable* [4] with no impact on the solution(s). A value, b , for a CSP variable, V , is *fully interchangeable* with a value, c , for V iff: **1.** Every solution to the CSP which contains b remains a solution when c is substituted for b , and **2.** Every solution to the CSP which contains c remains a solution when b is substituted for c . Where it is possible to eliminate interchangeable values in CSPs, a clustering of subsets of the original CSP variables may be achieved, with a set of *meta-variables* created.

A *meta-CSP* of a ground CSP, X , consists of variables that correspond to subsets of the variables in X . The values of the meta-variables are the solutions of the problems induced by the subsets of variables. The constraints between the meta-variables are satisfied when all of the constraints from the original CSP are satisfied. Applying this concept to a CSP encoded planning problem [5] allows a meta-CSP to be constructed, the variables of which represent the goals of the original planning problem. For a planning problem, $P = (O, I, G)$, the meta-CSP is a CSP where $|X| = |G|$ and $D_i = \{a \mid g_i \in \text{add}(a)\}$. Each variable represents a goal, $g_i \in G$, with the associated domain containing only the actions that achieve g_i . Such actions have g_i in the add list of their effects. With a variable assigned a value in this representation, $\langle x_i, a \rangle$, the meaning is that a is the *final achiever* of g_i . This means that it is not possible for any action appearing later in the plan than a to delete g_i . This is shown in the partial solution matrix below (Figure 2) for the example in Section 2.

It is clear from Figure 2 that the domain (search space) of each the action variables, Ac8 to Ac23, has been reduced, giving the heuristic algorithm even less choice than before, leading to more efficient solving (Figures 3 and 4).

The second use of meta variables allows a meta variable to be assigned a value depending on which of the available *resources* are allocated to each of a problem's *tasks*. By labelling these meta variables, and using this to direct the search for actions to form a plan to satisfy the problem's goals (tasks), it is possible to show that such plans can be built faster, often with fewer backtracks, and sometimes with fewer actions. Although this part of the project is ongoing, results look promising.

3	5	2	...	Ac1:	1: boardtruckdrivertruck1a0
3	0	2	...	Ac2:	22: drivetruck1a01a1driver1
3	0	2	...	Ac3:	44: loadtruckpackagetruck1a1
3	0	4	...	Ac4:	51: drivetruck1a11a2driver1
3	0	4	...	Ac5:	68: unloadtruckpackagetruck1a2
3	0	0	...	Ac6:	[0,6,8,9,12,15,16,19,22,31,...]
[0,2,7]	[0,3]	0	...	Ac7:	[0,2,4,8,9,12,15,16,19,22,31,...]
[0,4,7]	[0,2,6]	0	...	Ac8:	[1,2,4,10,12,15,43,45,53,55,56,...]
[0,7]	[0,7]	0	...	Ac9:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac10:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac11:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac12:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac13:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac14:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac15:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac16:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac17:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac18:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac19:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac20:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac21:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac22:	[1,2,4,10,12,43,45,66,70,75,78,80,...]
[0,7]	[0,7]	0	...	Ac23:	[1,2,4,10,13,34,...]
[0,7]	[0,7]	0	...	Ac24:	[5,8,13,21,23,26,...]
[0,7]	[0,7]	0	...	Ac25:	[5,6,13,20,26,27,...]
[1,4,6]	[1,5]	0	...	Ac26:	[5,6,14,16,26,27,...]
[1,2,4,6]	[1,2,4,3]	0	...	Ac27:	[1,4,16,58,59,68,...]
4	4	0	...		

Figure 2: Partial solution matrix with goal locking applied.

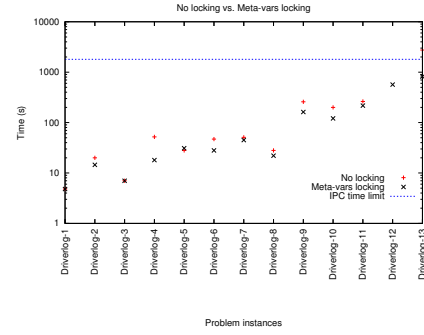


Figure 3: Runtime with & without meta-variables.

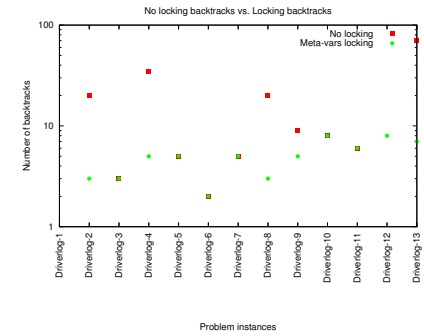


Figure 4: Backtracks with & without meta-variables.

4 Conclusion

Depending on the structure of a particular problem, action pruning resulting from the increased inferences (due to the use of meta variables) may lead to a reduction in the amount of both runtime and backtracking required to find a solution to the given problem. However, whilst such propagation may aid faster solution, the increased number of constraints and associated processing may be an overhead. Further, the overall impact will depend on the causal dependencies between as yet unassigned CSP variables and those recently locked.

References

- [1] Christer Backstrom. Equivalence and Tractability Results for SAS+ Planning. In *Proceedings of KR'92*. Morgan Kaufmann, 1992.
- [2] Rina Dechter. *Constraint processing*. Elsevier, 2003.
- [3] R. E. Fikes and N. J. Nilsson. Strips. In *Readings in Planning*, pages 88–97. Kaufmann, San Mateo, CA, 1990.
- [4] Eugene C. Freuder. Eliminating interchangeable values in cpsps. In *Procs of the 9th National Conference on AI - Volume 1*, 1991.
- [5] Peter Gregory, Derek Long, and Maria Fox. A Meta-CSP Model for Optimal Planning. In *Proceedings of Abstraction, Reformulation and Approximation, Seventh International Symposium*, 2007.
- [6] Malte Helmert and Silvia Richter. Fast Downward. In *Proceedings of the International Planning Competition 2004*, pages 41–43, 2004.
- [7] Mark Judge. Constraint-based Heuristic Guidance for Solution of AI Planning Problems. In *Procs of International IEEE / EPSRC Workshop on Autonomous Cognitive Robotics. Univ of Stirling*, 2014.
- [8] Mark Judge and Derek Long. A CSP Heuristic for AI Planning. In *Procs of 20th Automated Reasoning Workshop, School of Computing, University of Dundee, UK*, 2013.
- [9] Craig Knoblock. Pddl. *AIPS98*, 78(4):1–27, 1998.