A Compilation Based Approach to Finding Centroids and Minimum Covering States in Planning

Erez Karpas, Faculty of Industrial Engineering and Management, Technion, Israel

Motivation

Suppose we have a set of possible goals. One of these goals will “arrive” later, but we now have time to prepare for it.

We should go to either:

centroid a state that minimizes the average distance to each possible goal

minimum covering state that minimizes the maximum distance to each possible goal

Problem first presented by Pozanco et. al. [1]

Problem Statement

Given a STRIPS with multiple goals instance $\Pi = \langle F, A, I, \mathcal{G}, C\rangle$, where:

- F is a set of facts describing the possible states of the world, 2^F
- A is a set of actions; each action $a \in A$ is $\langle pre(a), add(a), del(a)\rangle$ with cost $C(a)$
- $I \subseteq F$ is the initial state of the world, and
- $\mathcal{G}$ is a set of possible goals, where each possible goal $G \in \mathcal{G}$ is a set of facts $G \subseteq F$. A state s satisfies a goal if $G \subseteq s$

Denote by $h^*(s, G)$ the cost of an optimal path from state s to a state s' such that $G \subseteq s'$

State s is a **centroid** iff: s is reachable from I, and $\sum_{i=1}^{n} h^*(s, G_i)$ is minimal (equivalent to minimizing average distance)

State s is a **minimum covering state** iff: s is reachable from I, and $\max_{i=1}^{n} h^*(s, G_i)$ is minimal

Solution: Intuition

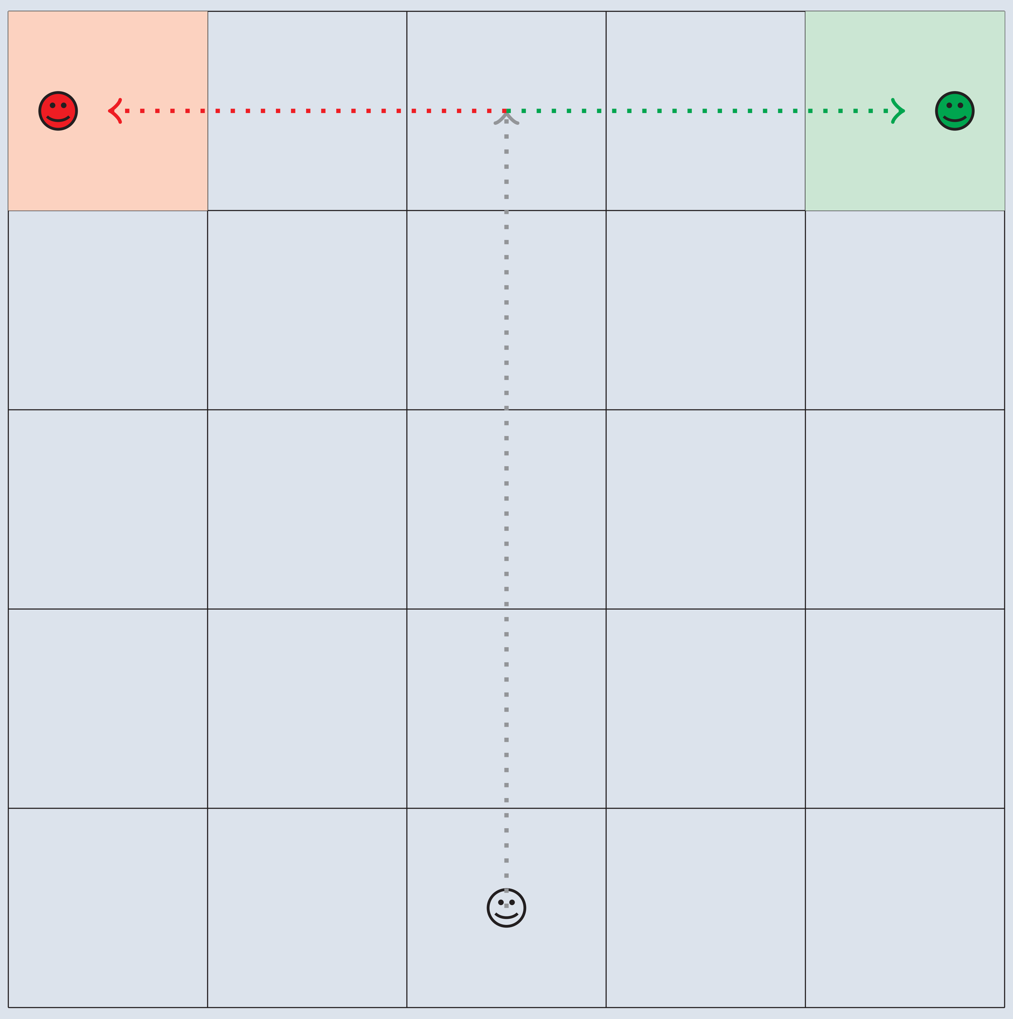
The problem statement is very similar to finding worst case distinctiveness (wcd) in Goal Recognition Design (GRD) [2]

Reminder: the wcd is the maximal number of steps an agent can take from the initial state before its goal becomes clear

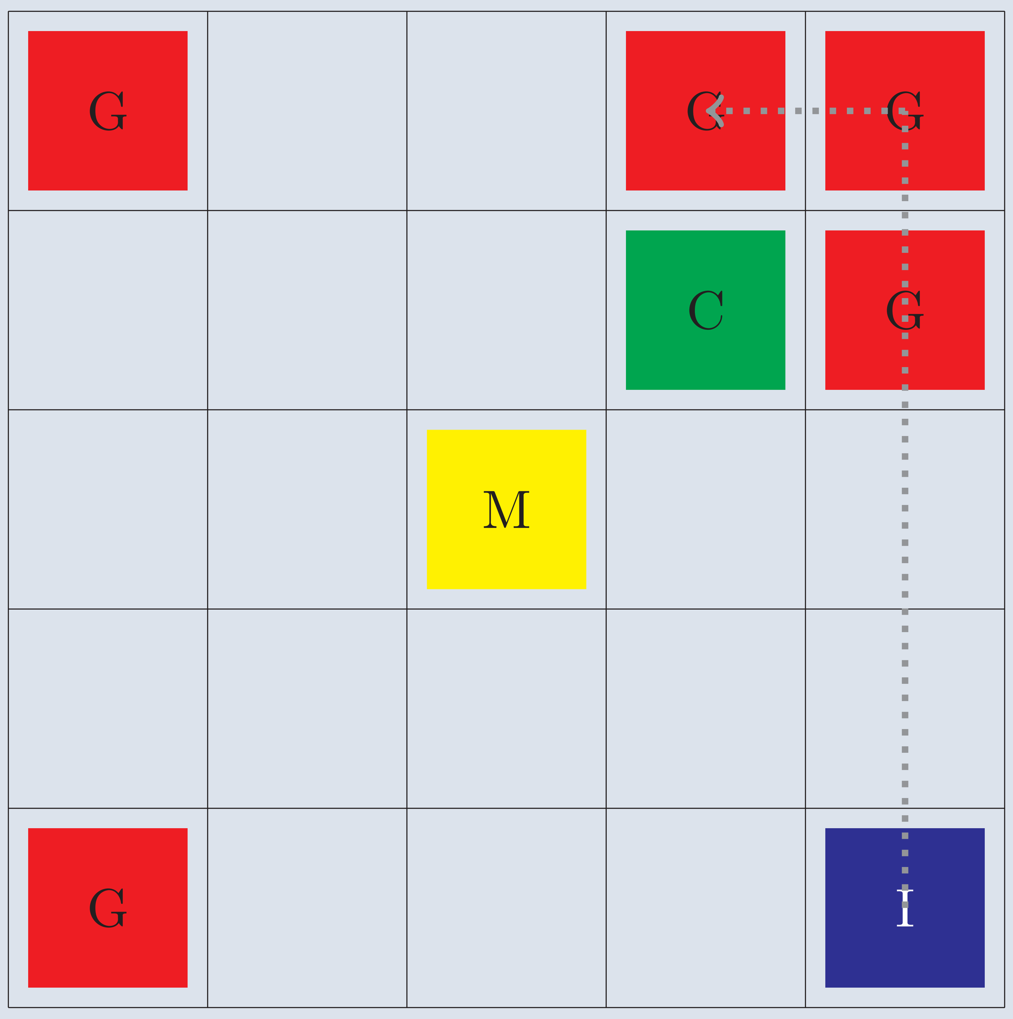
Finding wcd is done via compilation to classical planning

It turns out, the compilation for finding centroid states is very similar

Compilation



WCD $\neq$ Centroid $\neq$ Min-cover



Centroid Compilation

Create n copies of each fact (one for each state)

Create $n + 1$ copies of each action: a_t for all agents performing action a together, and a_i for agent i performing a alone

The only difference between the wcd compilation and this compilation are the costs:
$C(a_t) = 0$
$C(a_i) = C(a)$

The compilation finds paths from the initial state to all goals. The cost of a plan for the compilation is the sum of costs after splitting, thus the state where it splits is a centroid.

Min Cover Compilation

The max operator in minimum covering states is not additive. Instead, create a compilation with numeric variables that checks if there exists some reachable state s such that $\max_{G\in\mathcal{G}}, h^*(s, G) \leq B$.

Add n new numerical variables, $B_1 \dots B_n$. The value of B_i in the initial state is 0.

$B_i < B - C(a_i)$ is added to $pre(a_i)$, and $B_i += C(a_i)$ to the effects of a_i.

To optimize, perform binary search over B.

Similar to WCD with bounded deception cost [3]

Min Cover Unit Cost Actions

In the special case when all actions are unit cost, we can find the min cover state directly.

After splitting agents take turns executing actions in a round robin manner (without the optimization for enforcing the order between the agents)

Add $turn_i$ for turn taking, with appropriate preconditions/effects. Add NOOP actions, one for each agent, to allow agents to wait after reaching their goal.

The costs of actions are 1 for actions of agent 1 after splitting, 0 for all others (agent 1 is guaranteed to act in every round)

Results

	Centroid			Minimum Covering			
Domain	C	E	Sp	Cd	Cb	E	Sp
blks-w	10	10	41	10	10	10	7
fry	**10**	0	-	**10**	**10**	0	-
grpr	**10**	2	742	**10**	**10**	2	750
hanoi	**10**	6	373	**10**	**10**	6	355
log	**10**	2	195	**10**	**10**	2	189
IPC	**50**	20	226	**50**	**50**	20	204
g 5%	**10**	7	56	0	0	**7**	-
g 10%	**10**	8	92	1	0	**7**	0.2
g 15%	10	10	93	0	1	**10**	-
g 20%	**10**	9	74	0	0	**9**	-
grid	**40**	34	80	1	1	**33**	0.2
TOTAL	**90**	54	134	51	51	**53**	194

Conclusions

We presented a compilation based approach to finding centroids and minimum covering states

On IPC domains, compilation based approach is about 200X faster than baseline

On Grid finding centroids using compilation is 80X faster

Finding min cover states using compilation is much slower – due to the small size of the state space

References

[1] A. Pozanco, Y. E-Martin, S. Fernadez, and D. Borrajo, “Finding centroids and minimum covering states in planning,” in *ICAPS 2019*, AAAI Press, 2019, pp. 348–352.

[2] S. Keren, A. Gal, and E. Karpas, “Goal recognition design,” in *ICAPS*, AAAI, 2014.

[3] ——, “Goal recognition design for non-optimal agents,” in *AAAI*, AAAI Press, 2015, pp. 3298–3304.