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

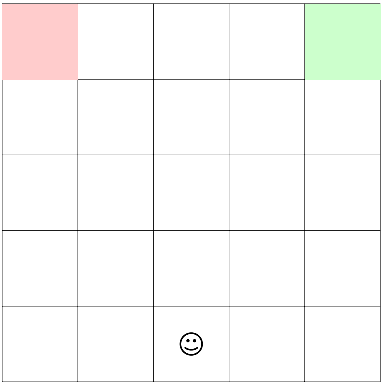
Erez Karpas

Technion — Israel Institute of Technology

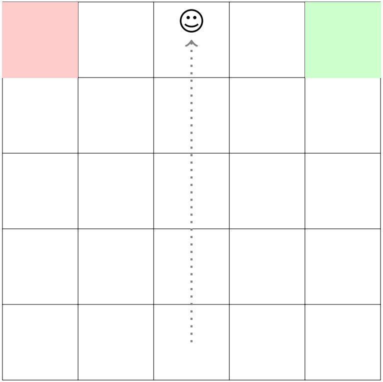
Motivation

- 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:
  - a centroid state - one that minimizes the average distance to each possible goal
  - a minimum covering state - one that minimizes the maximum distance to each possible goal
- Problem was first presented by Pozanco et. al. [PEFB19]

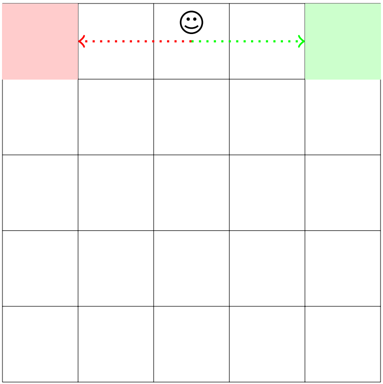
Example



Example



Example



Problem Setting

The setting here is STRIPS with multiple possible goals. Formally, $\Pi = \langle F, A, I, \mathscr{G}, C \rangle$, 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
- $\mathscr{G}$ is a set of possible goals, where each possible goal $G \in \mathscr{G}$ is a set of facts $G \subseteq F$. A state s satisfies a goal if $G \subseteq s$

Problem Objective

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

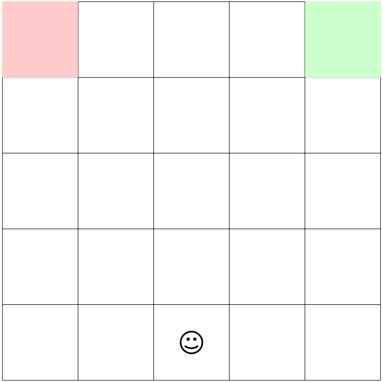
- State s is a centroid iff: s is reachable from I, and $\sum_{i=1}^{n} h^*(s, G_i)$ is minimal (equivalent to minimizing average distance)
- State s is a minimum covering state iff: s is reachable from I, and $\max_{i=1}^{n} h^*(s, G_i)$ is minimal

The objective is to find either a centroid or a minimum covering state, possibly also optimizing over the cost to get there

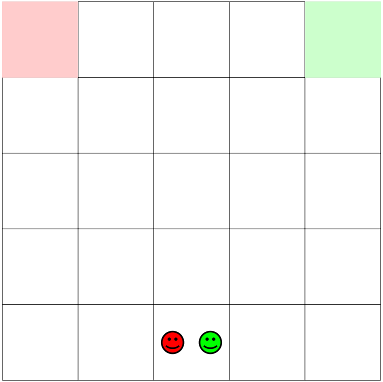
Inspiration

- The problem statement (and example) are very similar to finding worst case distinctiveness (wcd) in Goal Recognition Design (GRD) [KGK14]
- 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

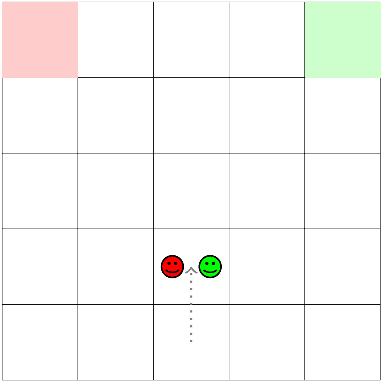
Compilation: Illustrated



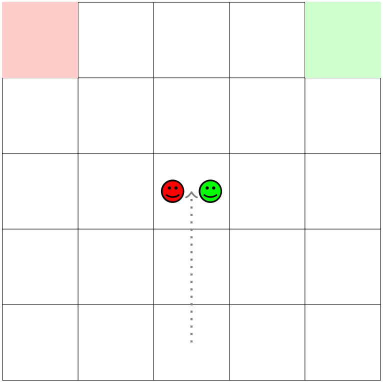
Compilation: Illustrated



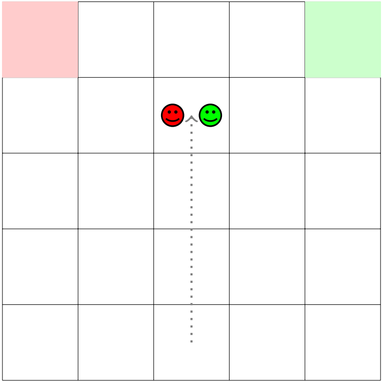
Compilation: Illustrated



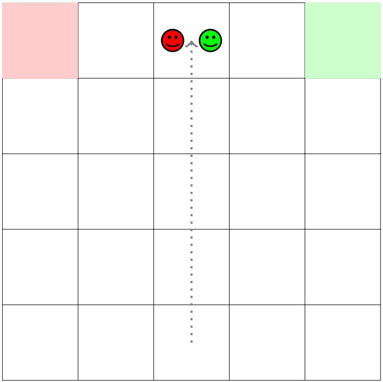
Compilation: Illustrated



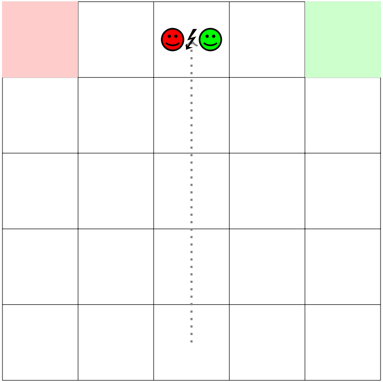
Compilation: Illustrated



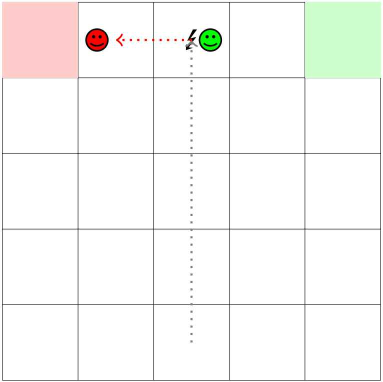
Compilation: Illustrated



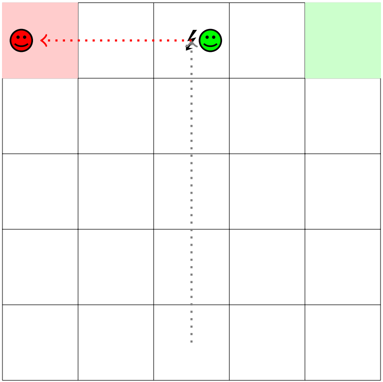
Compilation: Illustrated



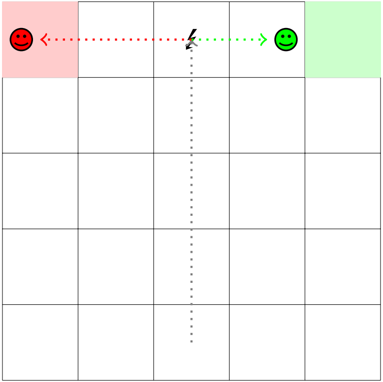
Compilation: Illustrated



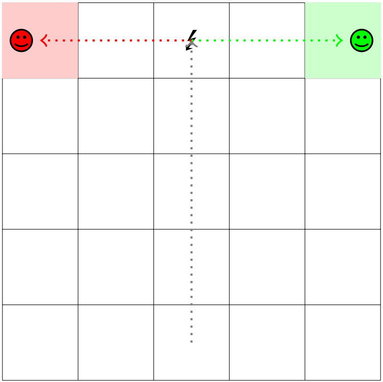
Compilation: Illustrated



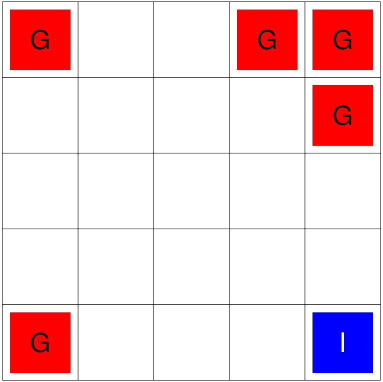
Compilation: Illustrated



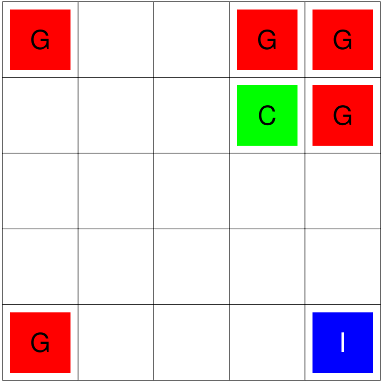
Compilation: Illustrated



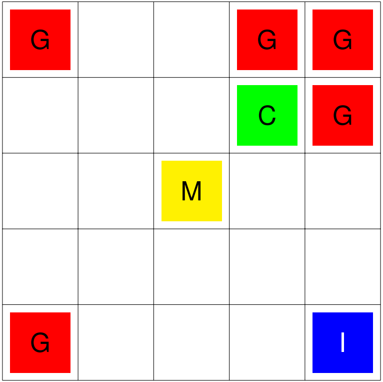
Caveat: WCD $\neq$ Centroid $\neq$ Min-cover



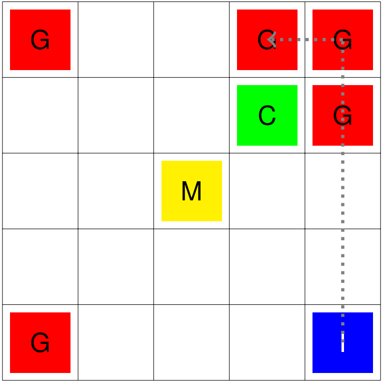
Caveat: WCD $\neq$ Centroid $\neq$ Min-cover



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



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



Centroid Compilation

Given $\Pi = \langle F, A, I, \mathscr{G} = \{G_1, \ldots G_n\}, C\rangle$ we define $\Pi' = \langle F', A', I', G', C'\rangle$, where:

- $F' = \{f_i \mid f \in F, i = 1 \ldots n\} \cup \{\text{split}, \text{unsplit}\}$,
- $A' = \{a_i \mid a \in A, i = 1 \ldots n\} \cup \{a_t \mid a \in A\} \cup \{\text{do-split}\}$, where
  - a_t is the together version of action a, affecting all of the f_i facts, and is possible only before splitting
  - a_i is the separate version of action a for goal i, affecting only the f_i variables, and is only possible after splitting
  - The do-split action allows the agents to split
- $I' = \{f_i \mid f \in I, i = 1 \ldots n\} \cup \{\text{unsplit}\}$
- $G' = \{f_i \mid f \in G_i, i = 1 \ldots n\}$

Centroid Compilation vs. wcd Compilation

The only difference between the wcd compilation and this compilation are the costs:

- In wcd, we want to maximize the costs of the "together" actions, so the costs are
  - $C(a_t) = nC(a) - \varepsilon$
  - $C(a_i) = C(a)$
- In finding centroids, we only care about the costs of the "separate" actions, so the costs are
  - $C(a_t) = 0$
    - If we want the compilation to find an optimal path to the centroid, we can set $C(a_t) = \varepsilon$ for a small enough ε
  - $C(a_i) = C(a)$
- In all cases $C(\text{do-split}) = 0$

Centroid Compilation: Theoretical Results

Theorem

An optimal solution for Π' gives us a centroid state for the original task Π.

Proof sketch.

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. □

Centroid Compilation: Optimizations

- We can force the agents to act in order after splitting – first agent 1 (until it reaches its goal), then agent 2, ...
- This reduces permutations of essentially the same plans

Finding Minimum Covering States

- Unfortunately, the max operator in minimum covering states is not additive
- Thus, we do not have a compilation which directly finds a minimum covering state in the general case
- We present a compilation which, given some cost budget B, checks whether there is some reachable state s such that the maximum cost of reaching any possible goal $G_i \in \mathscr{G}$ from s is at most B
- An binary search over B will find minimum covering states (starting by doubling B until the compilation is solvable)
- This is similar to the compilation for finding the wcd with non-optimal agents with deception budget [KGK15]

Minimum Covering Compilation: Version 1 (numeric)

The compilation is the same as the centroid compilation, except

- We 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
- Note that we only care about the cost of reaching the goals after splitting, so a_t actions are unmodified

Theorem

Let Π' be a numerical planning task with budget B as described above. Then Π' is solvable iff there exists some reachable state s such that $\max_{G \in \mathscr{G}}, h^*(s, G) \leq B$.

Minimum Covering Compilation: Version 2 (unit cost actions)

- If all actions are unit cost, we can compile finding the minimum covering state to classical planning (without binary search)
- After splitting agents take turns executing actions in a round robin manner (without the optimization for enforcing the order between the agents)
- The compilation is implemented by:
  - Adding n new facts, turn_i for $i = 1 \ldots n$
  - For each a_i action, we add turn_i to $pre(a_i)$, $\text{turn}_{i+1 \mod n}$ to $add(a_i)$, and turn_i to $del(a_i)$
  - Adding NOOP actions – one for each agent, to allow agents to wait after reaching their goal
  - The costs actions are 1 for actions of agent 1 after splitting, 0 for all others (agent 1 is guaranteed to act in every round)

Empirical Evaluation

- We compared our compilation (C) to the exhaustive search approach (E) presented in the previous work
- Used several IPC domains and grid navigation with X% obstacles
- Underlying planner was the same in both cases: Fast Downward [Hel06] with A* [HNR68] and the lmcut heuristic [HD09]
- Time limit of 1 hour, memory limit of 16GB

Empirical Results

	Centroid			Minimum Covering			
Domain	C	E	Spdup	Cd	Cb	E	Spdup
blocks-w	10	10	41.25	10	10	10	7.10
ferry	**10**	0	-	**10**	**10**	0	-
gripper	**10**	2	741.59	**10**	**10**	2	749.91
hanoi	**10**	6	372.86	**10**	**10**	6	355.36
logistics	**10**	2	195.32	**10**	**10**	2	188.97
IPC	**50**	20	226.17	**50**	**50**	20	204.05
grid 5%	**10**	7	56.16	0	0	**7**	-
grid 10%	**10**	8	92.20	1	0	**7**	0.19
grid 15%	10	10	93.16	0	1	**10**	-
grid 20%	**10**	9	74.12	0	0	**9**	-
grid	**40**	34	80.27	1	1	**33**	0.19
TOTAL	**90**	54	134.31	51	51	**53**	194.34

Empirical Results: Takeaways

- 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

Conclusion

- We presented a compilation based approach to finding centroids and minimum covering states
- Empirical performance for centroids is state-of-the-art
- Empirical performance for minimum covering states varies

Thank You

Thank You

Questions?

References I

- Malte Helmert and Carmel Domshlak, *Landmarks, critical paths and abstractions: What's the difference anyway?*, ICAPS 2009, AAAI, 2009.
- Malte Helmert, *The fast downward planning system*, J. Artif. Intell. Res. **26** (2006), 191–246.
- Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, *A formal basis for the heuristic determination of minimum cost paths*, IEEE Transactions on Systems Science and Cybernetics **SSC-4(2)** (1968), 100–107.
- Sarah Keren, Avigdor Gal, and Erez Karpas, *Goal recognition design*, ICAPS, AAAI, 2014.
- ______, *Goal recognition design for non-optimal agents*, AAAI, AAAI Press, 2015, pp. 3298–3304.

References II

Alberto Pozanco, Yolanda E-Martín, Susana Fernández, and Daniel Borrajo, *Finding centroids and minimum covering states in planning*, ICAPS 2019, AAAI Press, 2019, pp. 348–352.