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

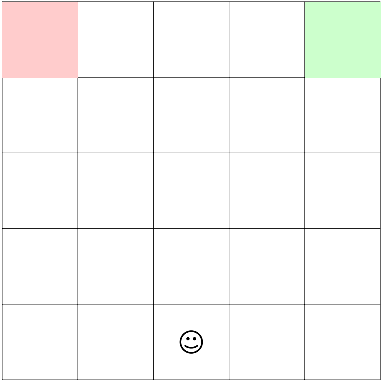
Erez Karpas

Faculty of Industrial Engineering and Management
Technion — Israel Institute of Technology

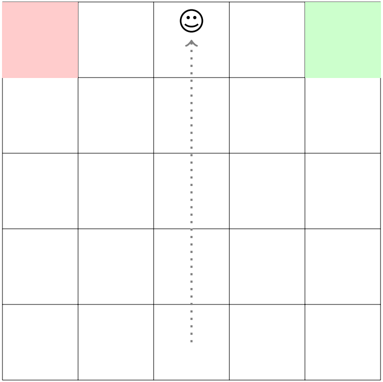
Motivation

- 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]

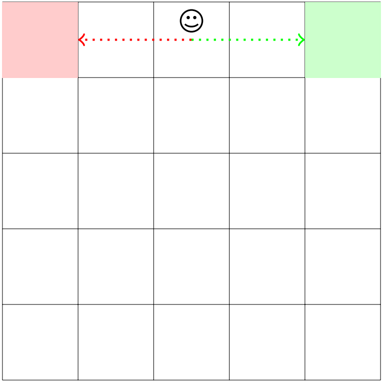
Example



Example



Example



Problem Setting

The setting here is STRIPS with multiple possible goals. Formally, $\Pi = \langle F, A, I, \mathcal{G}, C \rangle$, 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 $G \subseteq s$

Problem Objective

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

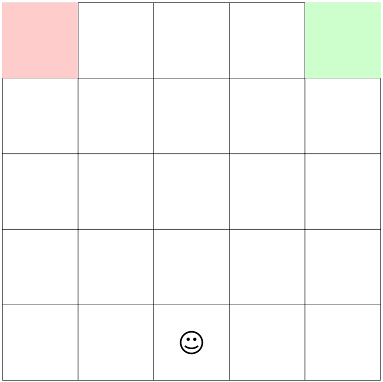
- State s is a centroid iff: s is reachable from I, and $\sum_{i=1}^{n} h^*(s, G_i)$ is minimal (equivalent to minimizing average distance)
- State s is a minimum covering state iff: s is reachable from I, and $\max_{i=1}^{n} h^*(s, G_i)$ is minimal

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

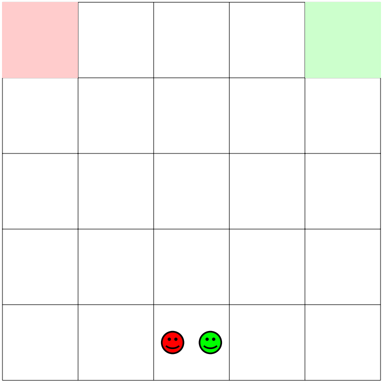
Inspiration

- 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

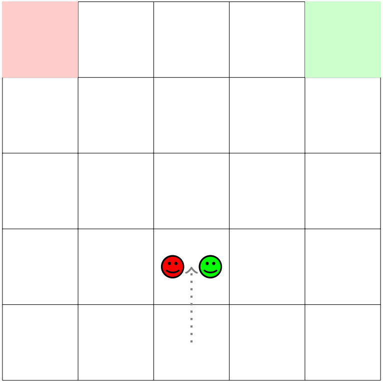
Compilation: Illustrated



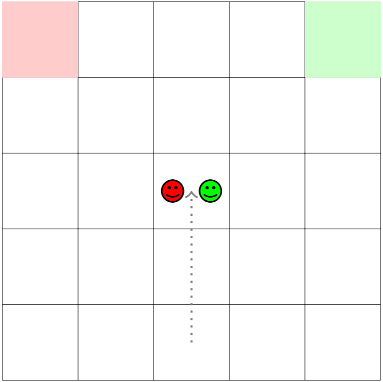
Compilation: Illustrated



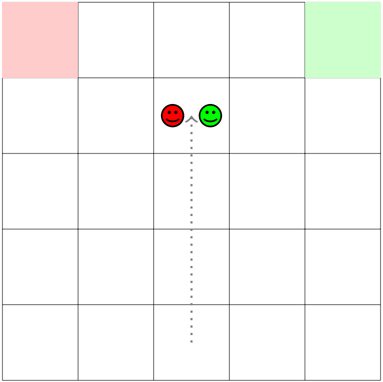
Compilation: Illustrated



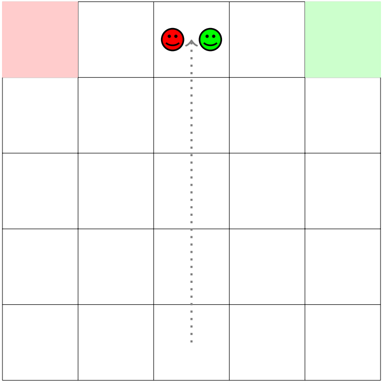
Compilation: Illustrated



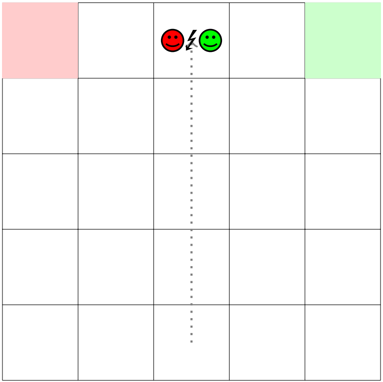
Compilation: Illustrated



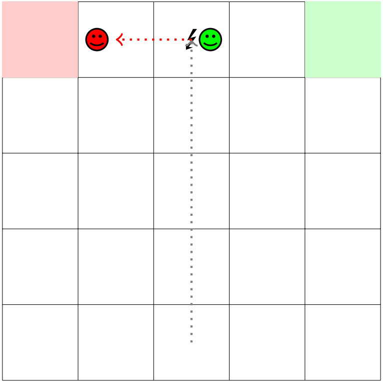
Compilation: Illustrated



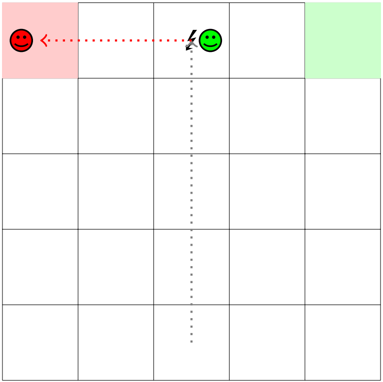
Compilation: Illustrated



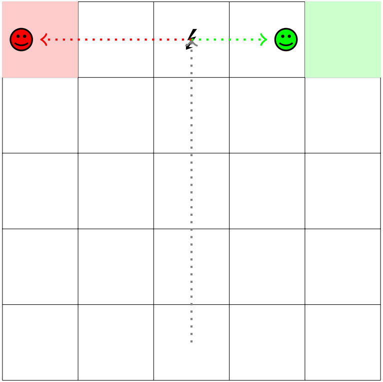
Compilation: Illustrated



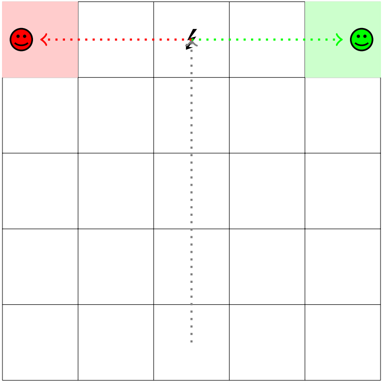
Compilation: Illustrated



Compilation: Illustrated



Compilation: Illustrated



Centroid Compilation

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

- $F' = \{f_i \mid f \in F, i = 1 \dots n\} \cup \{\text{split}, \text{unsplit}\}$,
- $A' = \{a_i \mid a \in A, i = 1 \dots n\} \cup \{a_t \mid a \in A\} \cup \{\text{do-split}\}$, 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
- $I' = \{f_i \mid f \in I, i = 1 \dots n\} \cup \{\text{unsplit}\}$
- $G' = \{f_i \mid f \in G_i, i = 1 \dots n\}$

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
  - $C(a_t) = 0$
  - $C(a_i) = C(a)$
- 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, $B_1 \ldots B_n$
- 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 $\max_{G \in \mathcal{G}}, h^*(s, G) \leq B$.

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 two domains from the previous work: BLOCKS-WORDS and RANGER (navigating on a grid)
- 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

	E	C
BLOCKS-WORDS		
1	838.04	25.89
2	835.30	16.05
3	852.02	11.03
4	821.71	22.96
5	818.97	7.18
6	835.56	15.58
7	810.36	9.06
8	818.74	8.69
9	827.43	16.09
10	830.51	12.18
AVG	828.86	14.47

	E	C
RANGER		
1	2197.42	14.31
2	2102.58	16.46
3	3124.19	17.16
4	1984.45	14.23
5	2140.93	16.59
6	1974.24	14.57
7	2126.50	17.13
8	2227.33	16.02
9	2128.61	14.31
10	2371.44	15.95
AVG	2237.77	15.67

Empirical Results: Minimum Covering

	E	Cd	Cb
BLOCKS-WORDS			
1	824.75	136.61	301.94
2	867.98	434.46	484.27
3	853.32	198.46	311.41
4	812.95	31.96	214.00
5	814.90	32.21	257.64
6	826.78	403.20	538.93
7	833.61	44.56	282.84
8	805.54	84.80	352.90
9	820.50	27.13	207.28
10	827.93	97.37	317.67
AVG	828.83	149.07	326.89

	E	Cd	Cb
RANGER			
1	2162.87	TO	TO
2	2118.05	TO	TO
3	2732.27	TO	TO
4	2042.48	TO	TO
5	2189.26	TO	TO
6	2031.07	TO	TO
7	2155.46	TO	TO
8	2169.37	TO	TO
9	2199.38	TO	TO
10	2368.51	TO	TO
AVG	2216.87	-	-

Cd = direct compilation, Cb = binary search using our compilation

Conclusion

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