LaCAM: Search-Based Algorithm for Quick Multi-Agent Pathfinding

Ali Czech

LaCam

- lazy constraints addition search
- 2 level search architecture
- Handles massive number of agents
- Sub-optimal
- Complete

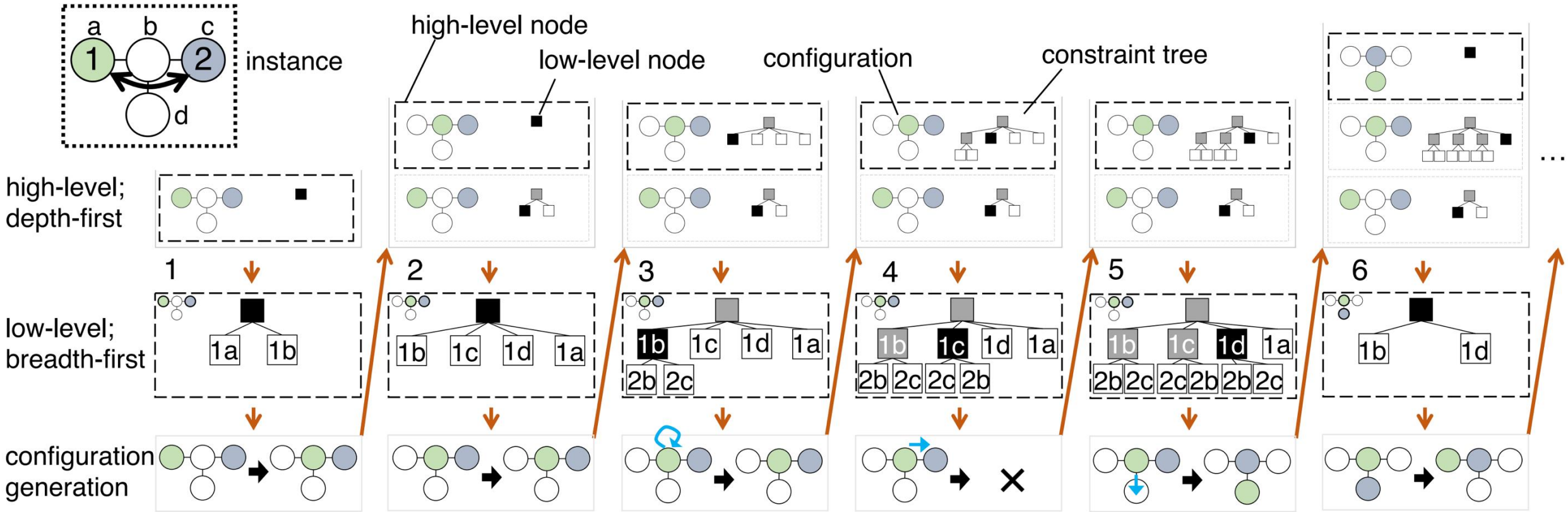


Figure 1: Running example of LaCAM. Orange arrows represent the search progress order. Selected and searched low-level nodes are filled with black and gray, respectively. Constraints are shown by blue-colored arrows.

Key Steps

- High-Level Search
- Low-Level Search
- Configuration Generation (PIBT)
- Discarding High-Level Nodes

Algorithm 1: LaCAM

input: MAPF instance ($\mathcal{S}$: starts, $\mathcal{G}$: goals)
output: solution or NO_SOLUTION
preface: $\mathcal{C}_{\text{init}} := \langle\, parent : \bot, who : \bot, where : \bot \,\rangle$

1: initialize $Open$, $Explored$ ▷ $Open$: *stack*
2: $\mathcal{N}_{\text{init}} \leftarrow \langle config : \mathcal{S}, tree : [\![\, \mathcal{C}_{\text{init}} \,]\!],$ ▷ $tree$: queue
$order : \mathsf{get\_init\_order}(), parent : \bot\rangle$
3: $Open.\mathsf{push}(\mathcal{N}_{\text{init}})$; $Explored[\mathcal{S}] = \mathcal{N}_{\text{init}}$
4: **while** $Open \neq \emptyset$ **do**
5: $\mathcal{N} \leftarrow Open.\mathsf{top}()$
6: **if** $\mathcal{N}.config = \mathcal{G}$ **then return** $\mathsf{backtrack}(\mathcal{N})$
7: **if** $\mathcal{N}.tree = \emptyset$ **then** $Open.\mathsf{pop}()$; **continue**
8: $\mathcal{C} \leftarrow \mathcal{N}.tree.\mathsf{pop}()$
9: **if** $\mathsf{depth}(\mathcal{C}) \leq |A|$ **then**
10: $i \leftarrow \mathcal{N}.order[\mathsf{depth}(\mathcal{C})]$; $v \leftarrow \mathcal{N}.config[i]$
11: **for** $u \in \mathsf{neigh}(v) \cup \{v\}$ **do**
12: $\mathcal{C}_{\text{new}} \leftarrow \langle\, parent : \mathcal{C}, who : i, where : u \,\rangle$
13: $\mathcal{N}.tree.\mathsf{push}(\mathcal{C}_{\text{new}})$
14: $\mathcal{Q}_{\text{new}} \leftarrow \mathsf{get\_new\_config}(\mathcal{N}, \mathcal{C})$
15: **if** $\mathcal{Q}_{\text{new}} = \bot$ **then continue**
16: **if** $Explored[\mathcal{Q}_{\text{new}}] \neq \bot$ **then continue**
17: $\mathcal{N}_{\text{new}} \leftarrow \langle config : \mathcal{Q}_{\text{new}}, tree : [\![\, \mathcal{C}_{\text{init}} \,]\!],$
$order : \mathsf{get\_order}(\mathcal{Q}_{\text{new}}, \mathcal{N}), parent : \mathcal{N}\rangle$
18: $Open.\mathsf{push}(\mathcal{N}_{\text{new}})$; $Explored[\mathcal{Q}_{\text{new}}] = \mathcal{N}_{\text{new}}$
19: **return** NO_SOLUTION

Theorem 1 (completeness)**.** *Algorithm 1 returns a solution for solvable MAPF instances; otherwise, it reports* `NO_SOLUTION`.

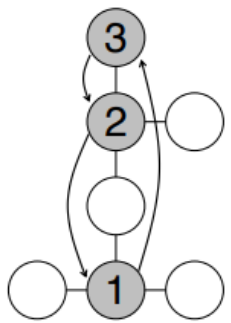
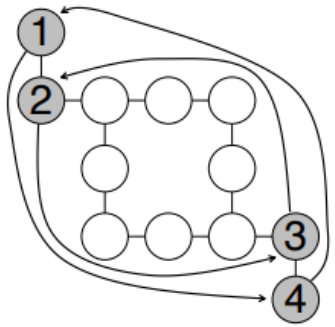
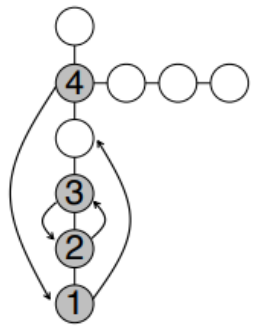
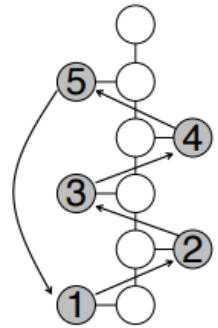
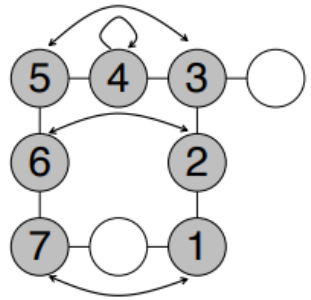
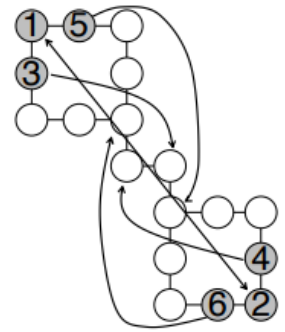
	tree		*corners*		*tunnel*		*string*		*loop-chain*		*connector*		
													
	Time(ms)	SOC	Time(ms)	SOC	Time(ms)	SOC	Time(ms)	SOC	Time(ms)	SOC	Time(ms)	SOC	Solved
LaCAM(med)	0	19	17	47	173	190	0	66	34	1752	0	168	**6/6**
LaCAM(worst)	0	41	31	70	208	254	0	68	124	2593	3	281	
PP	N/A	N/A	0	32	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1/6
OD	0	31	0	47	30	191	0	22	5882	2269	27	129	**6/6**
PIBT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0/6
PIBT$^+$	0	55	0	54	0	227	0	52	N/A	N/A	0	130	5/6
EECBS	1	16	1	32	N/A	N/A	0	20	N/A	N/A	N/A	N/A	3/6
LNS2	N/A	N/A	0	32	N/A	N/A	0	20	N/A	N/A	160	80	3/6
A*(SOC-optimal)	0	16	16	32	24	53	1	20	17752	121	391138	80	**6/6**

Table 1: Results of the small complicated instances.

warehouse-20-40-10-2-2

- Dimensions
  = 340x164
- # States
  = 38,756
- Corridor width
  = 2

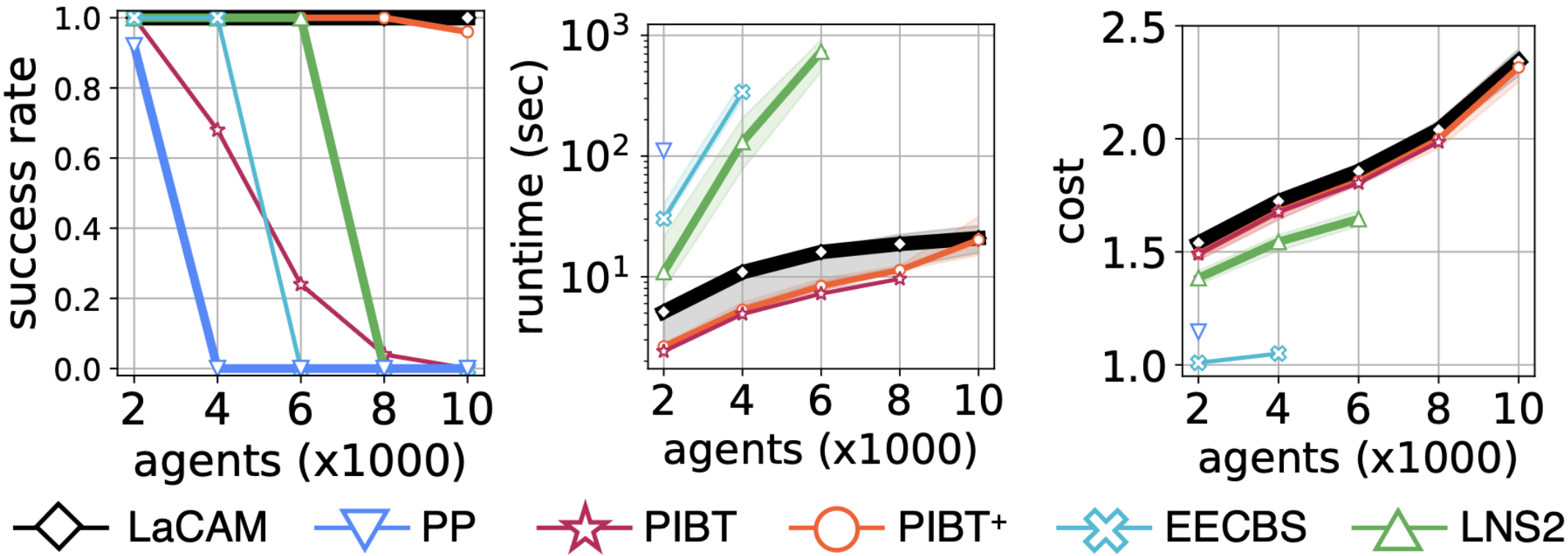
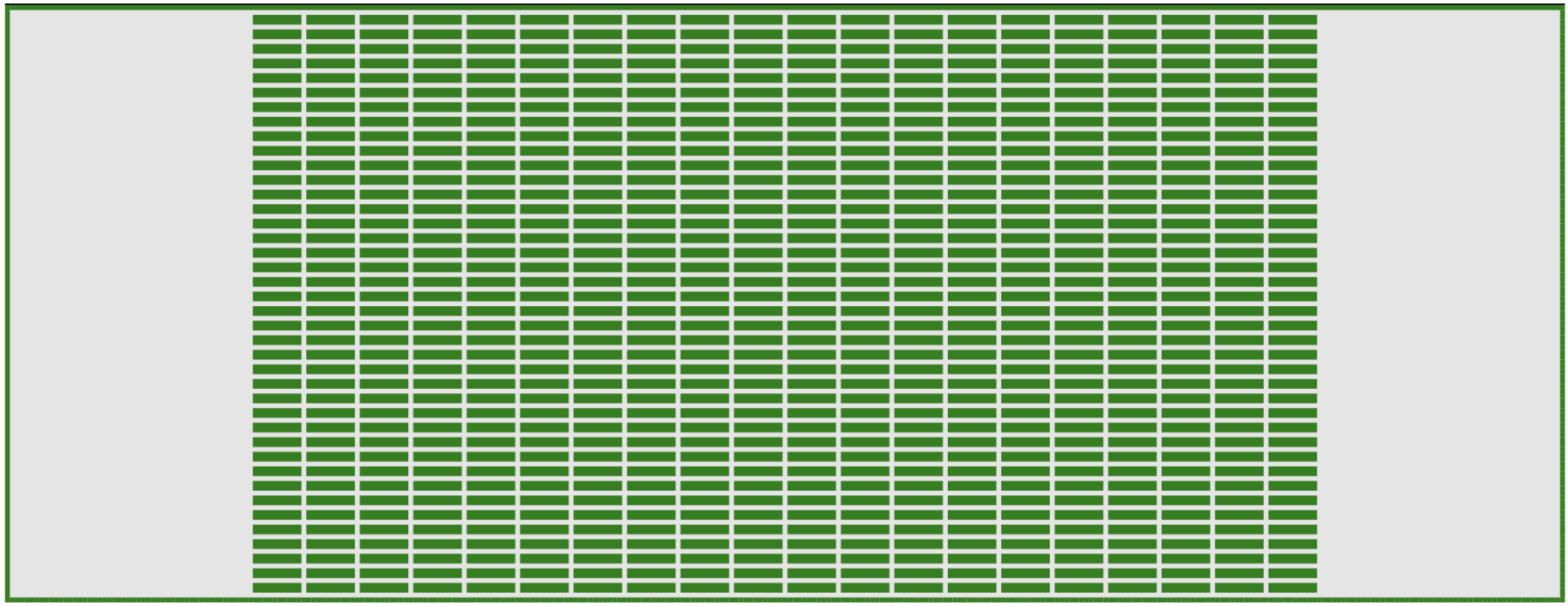
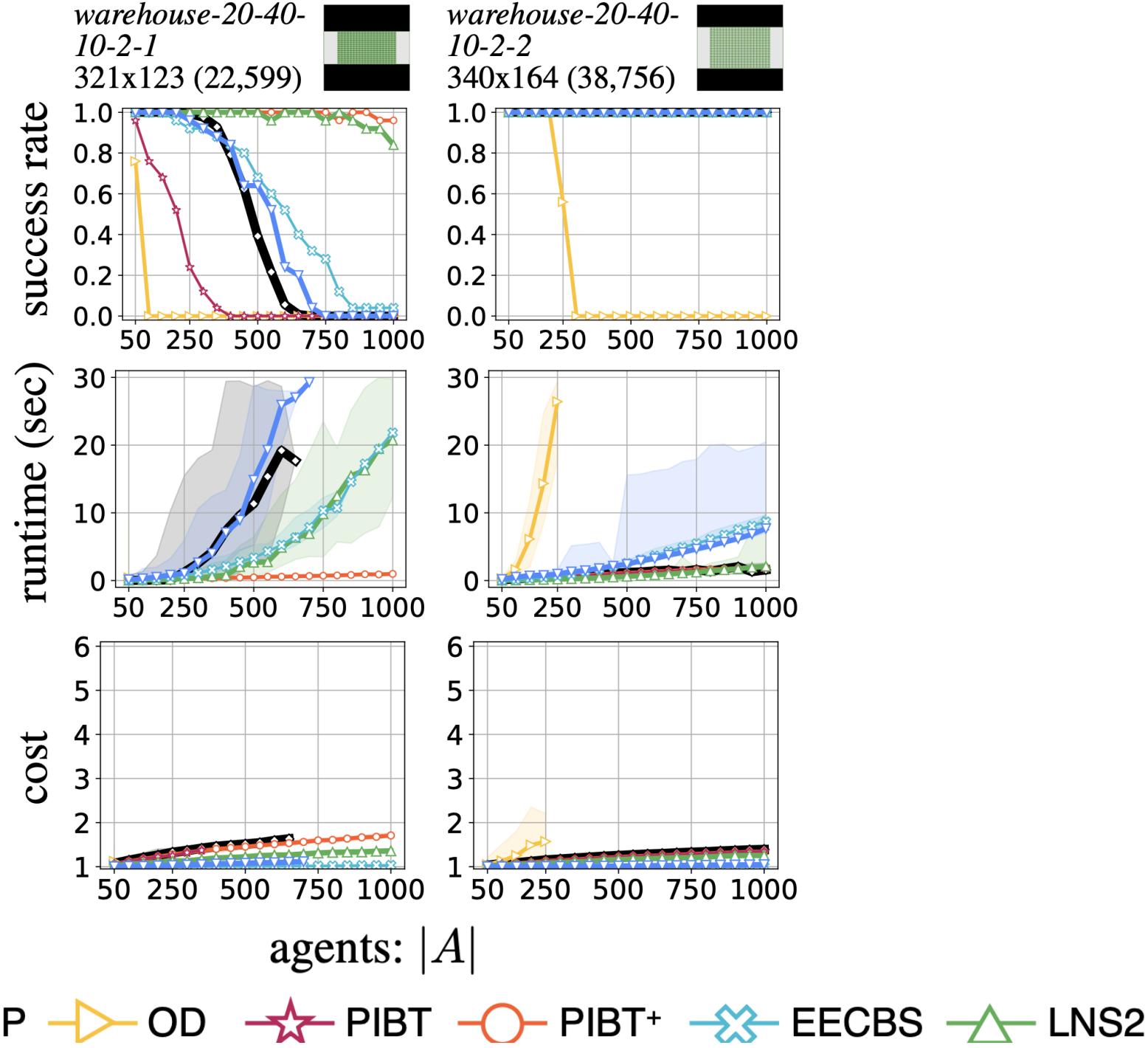


Figure 3: Results with massive agents. The used map was *warehouse-20-40-10-2-2*.

warehouse-20-40-10-2-1

- Dimensions
  = 321x123
- # States
  = 22,599
- Corridor width
  = 1




warehouse-20-40-10-2-1
321x123 (22,599)
warehouse-20-40-10-2-2
340x164 (38,756)
success rate
runtime (sec)
cost
agents: |A|
LaCAM
PP
OD
PIBT
PIBT+
EECBS
LNS2

Thank you!