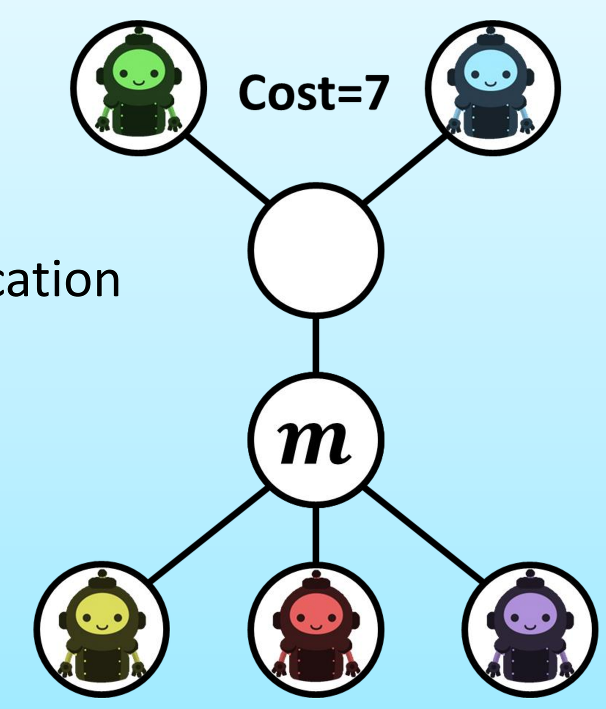
Conflict-Free Multi-Agent Meeting

Dor Atzmon, Shahar Idan Freiman, Oscar Epshtein, Oran Shichman, Ariel Felner
Ben-Gurion University of the Negev, Israel

Ben-Gurion University of the Negev

1. Multi-Agent Meeting (MAM)

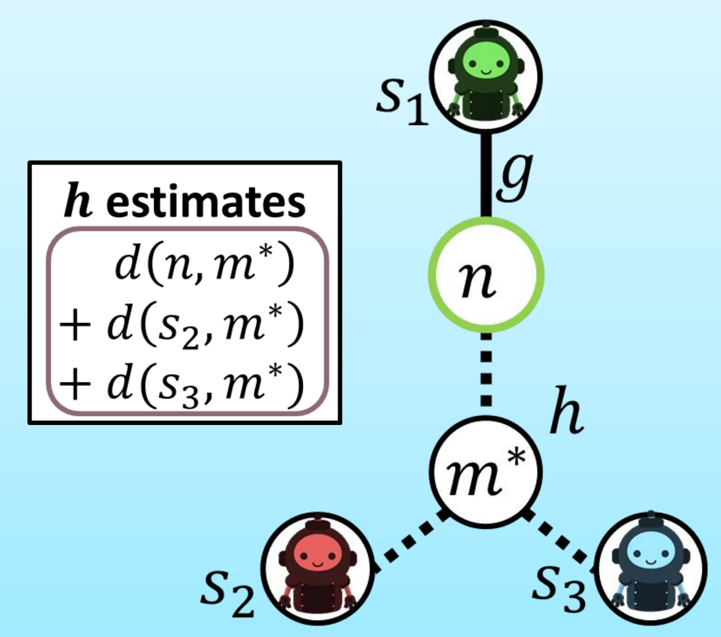
- **Input**
  - A map with N locations
  - A set of k agents, each with a start location
- **Actions** - An agent can move to an adjacent location
- **Task** - Find a *meeting location* ***m*** and a *path* for each agent to the meeting location
- **Target** - Minimize the sum of travel costs



2. Multi-Directional MM (MM*) for MAM [Atzmon et al. 2020]

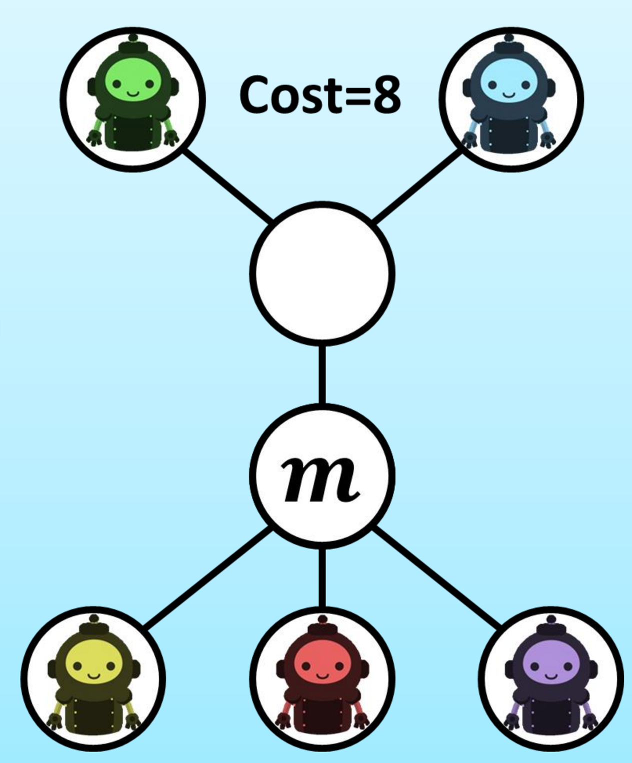
- Multi-directional heuristic search algorithm
- MM* progresses from all start locations
- MM* explores nodes by their f value
- $d(v_1, v_2)$ - Cost of the shortest path between v_1 and v_2

$$f(n) = g(n) + h(n)$$



3. Conflict-Free Multi-Agent Meeting (CF-MAM)

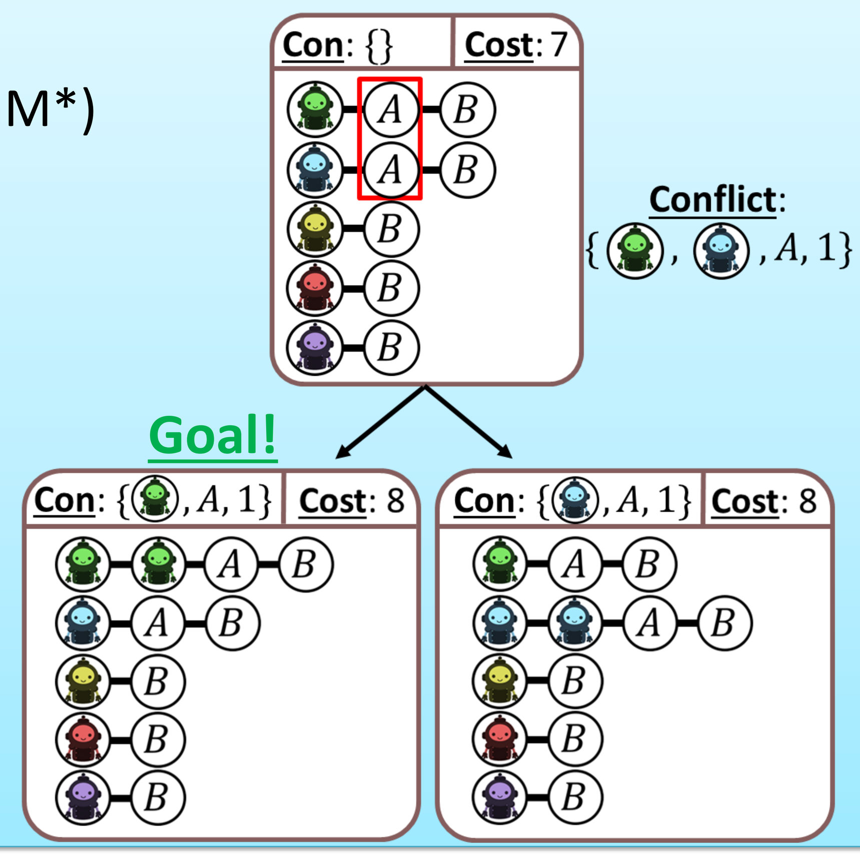
- **Input**
  - A map with N locations
  - A set of k agents, each with a start location
- **Actions** - An agent can **move or wait**
- **Task** - Find a *meeting location* ***m*** and a *path* for each agent to the meeting location
- **Constraints** - **Avoid conflicts**
- **Target** - Minimize the sum of travel costs

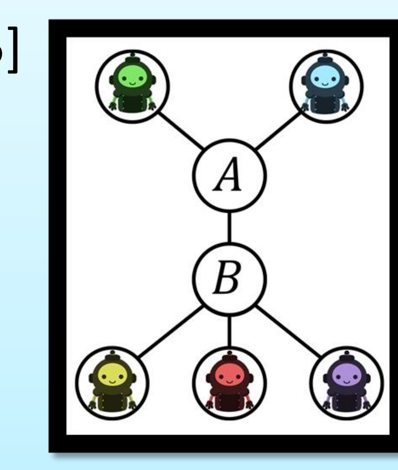


4. CFM-CBS for CF-MAM

- CFM-CBS uses the framework of CBS [Sharon et al. 2015]

1. Solve as MAM (MM*)
2. Identify conflicts
3. Set constraints to avoid the conflict
4. Replan (MM*)
5. Perform BFS

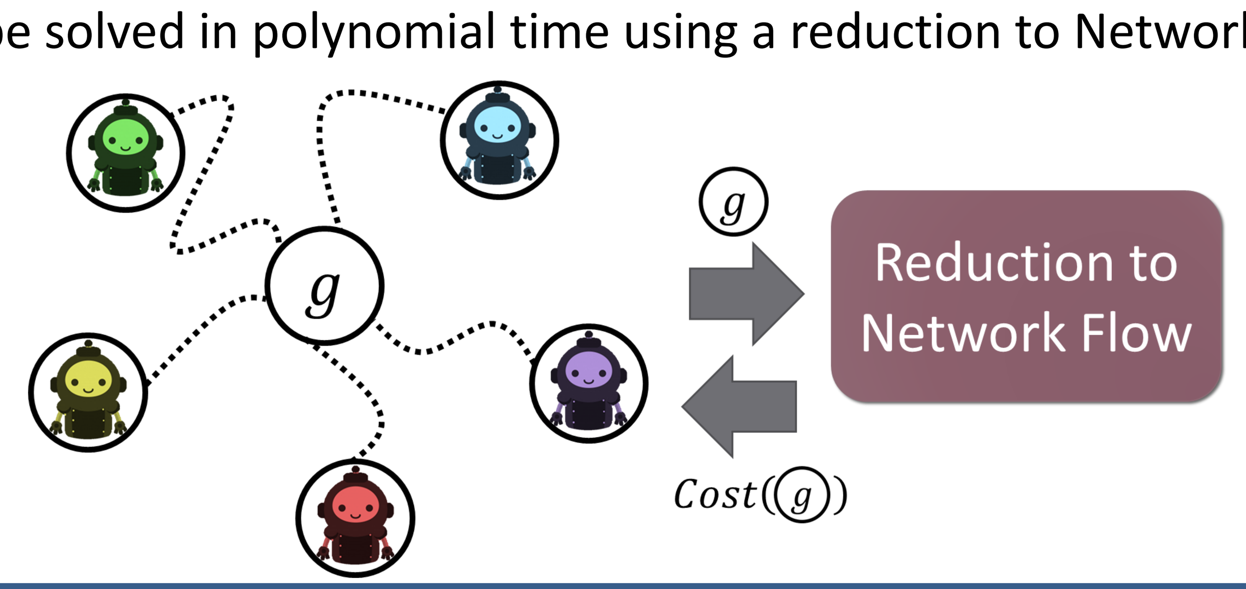




5. Iterative Meeting Search (IMS) for CF-MAM

Share-Goal Multi-Agent Path Finding (SG-MAPF)

- SG-MAPF searches conflict-free paths to a given location g
- Can be solved in polynomial time using a reduction to Network Flow



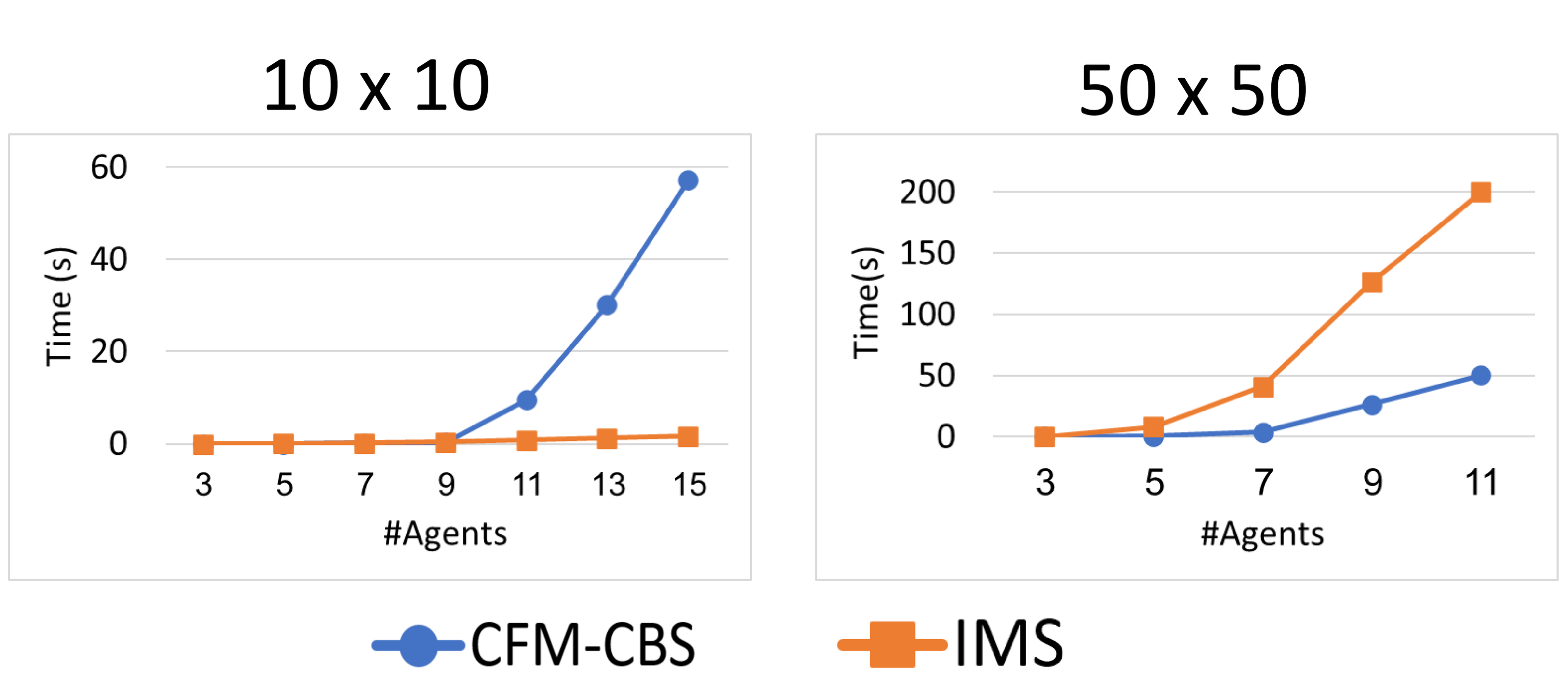
Iterative Meeting Search (IMS)

- IMS examines potential meeting locations using a similar reduction ←

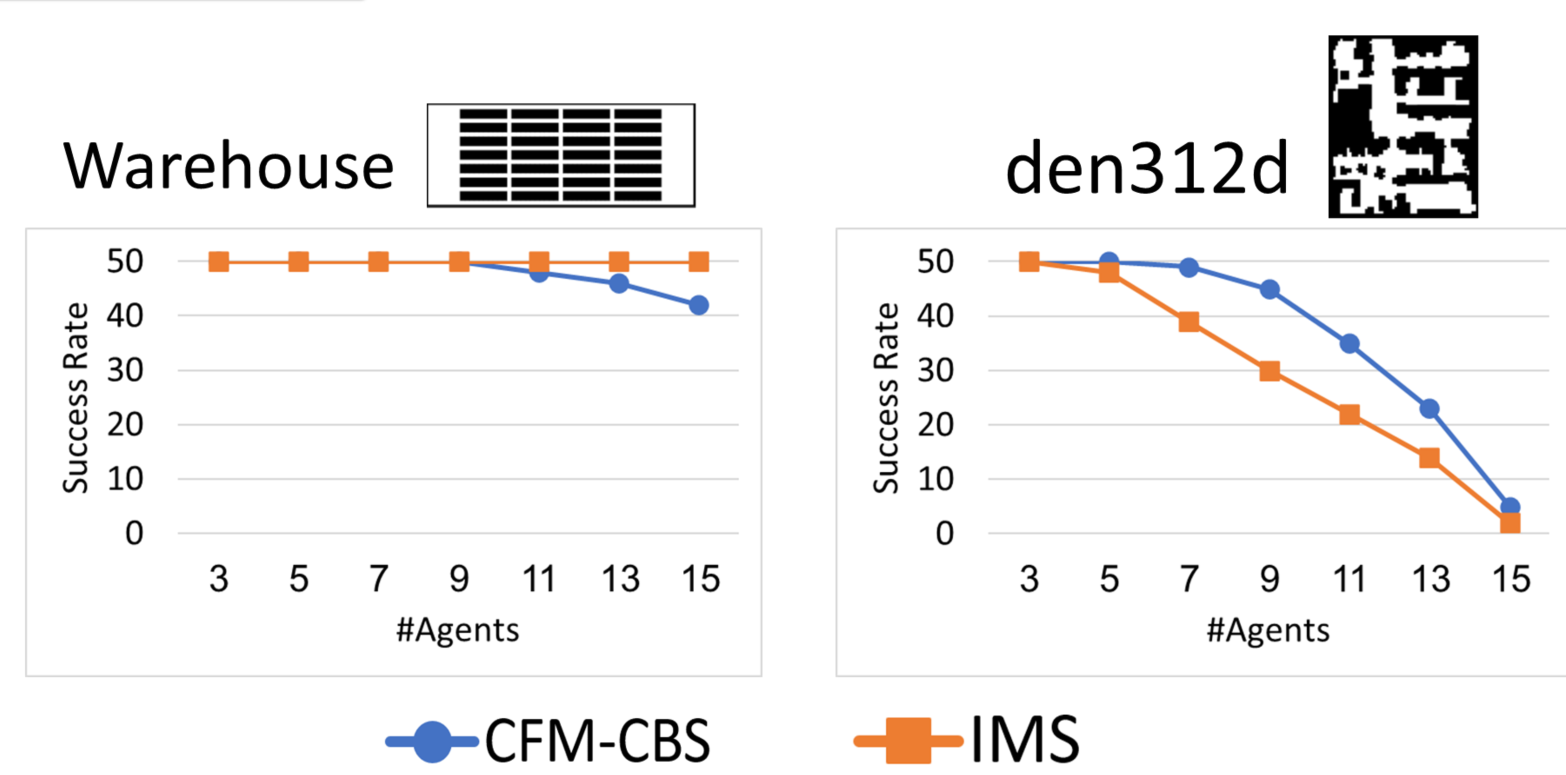
1. Insert one of the start locations into open (organized by the same f value as in MM*)
2. Extract from open the node n with the lowest f value and
3. Check the cost of meeting at this location using a reduction
4. Update the upper bound U with this cost
5. Insert n's successors into open and continue the same process
6. Halt if $fmin$ is larger than or equal to U

6. Experimental Results

Random grids (20% Obstacles)



Structured Maps



- For sparse domains, CFM-CBS is best
- For dense domains, IMS is best

7. Future Work

- Improve CFM-CBS by adjusting improvements for CBS
- Improve IMS by suggesting sophisticated rules for calling the reduction