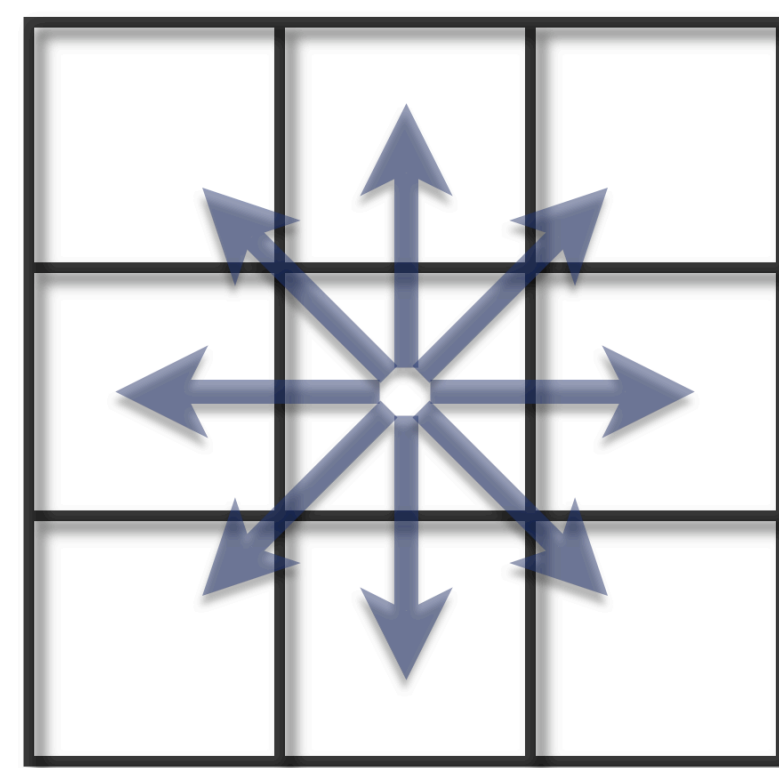


Biased Cost Pathfinding

Alborz Geramifard, Pirooz Chubak, Vadim Bulitko
University of Alberta

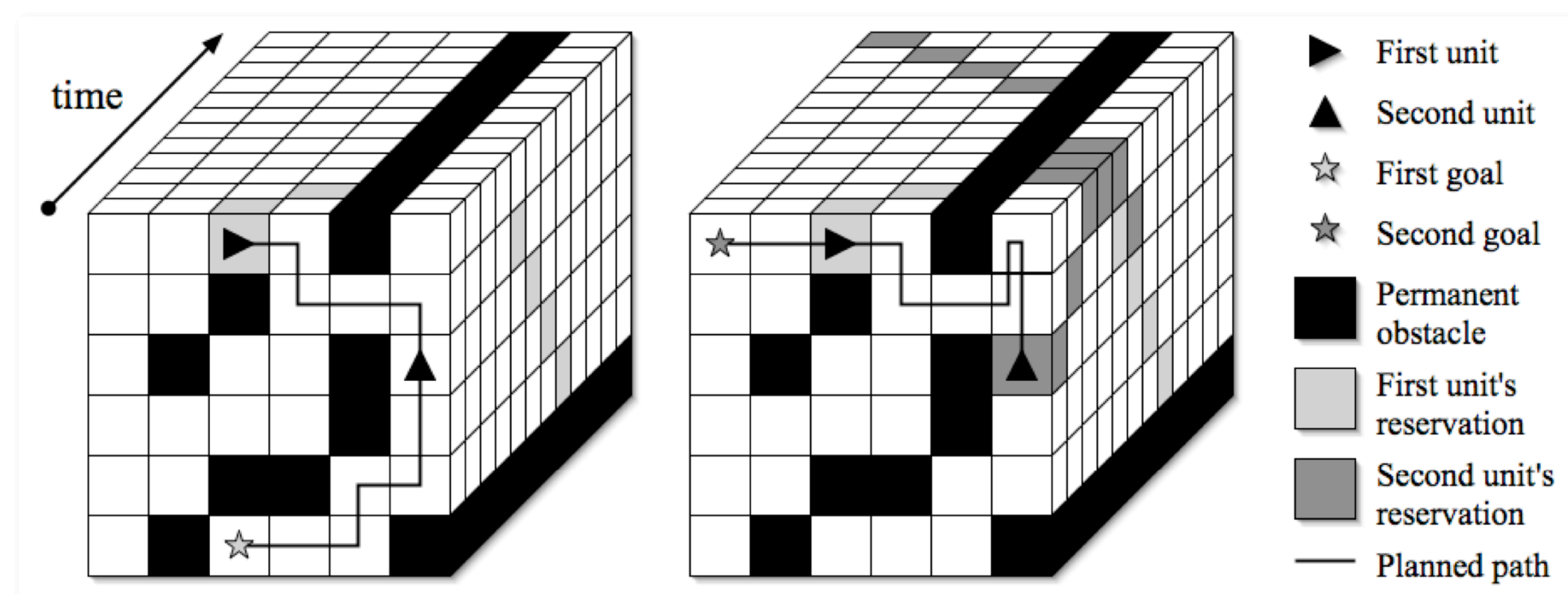
Problem

- Multi Agent Pathfinding
- Deterministic Environment (Grid World)
- Complete Information
- Identical Agents with Different Priorities
- Same Cost for Actions

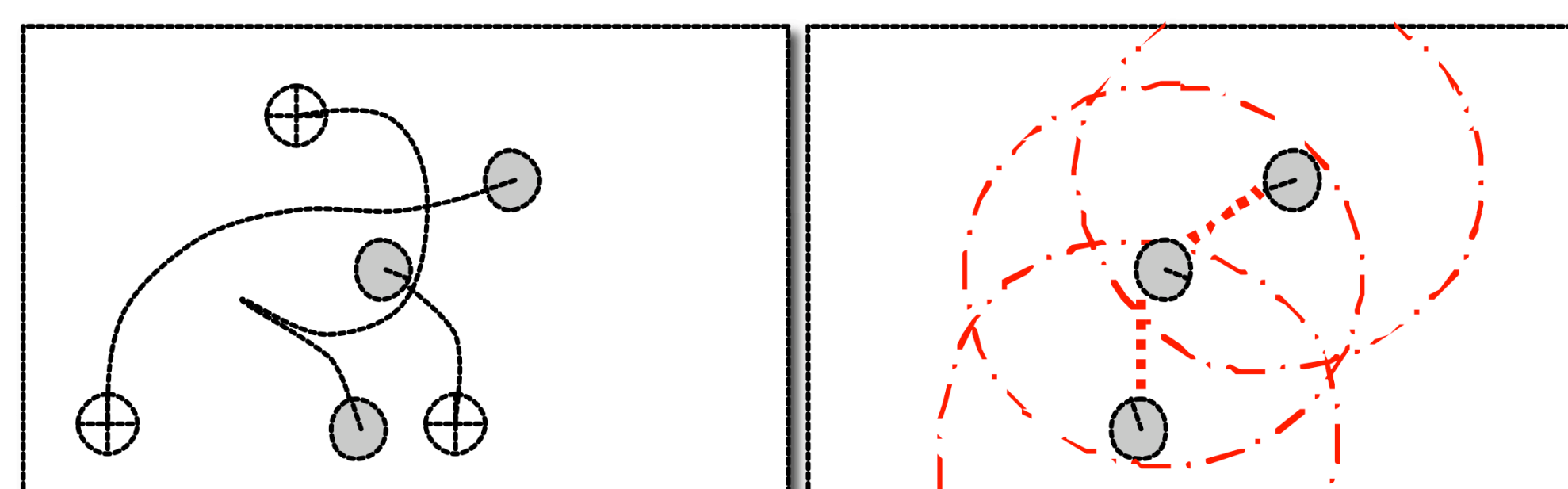


Related Works

- Cooperative Pathfinding, **Centralized** [Silver 2005, 2006]



- Collision Prevention on demand, **Decentralized** [Clark et al. 2003]



New Approach

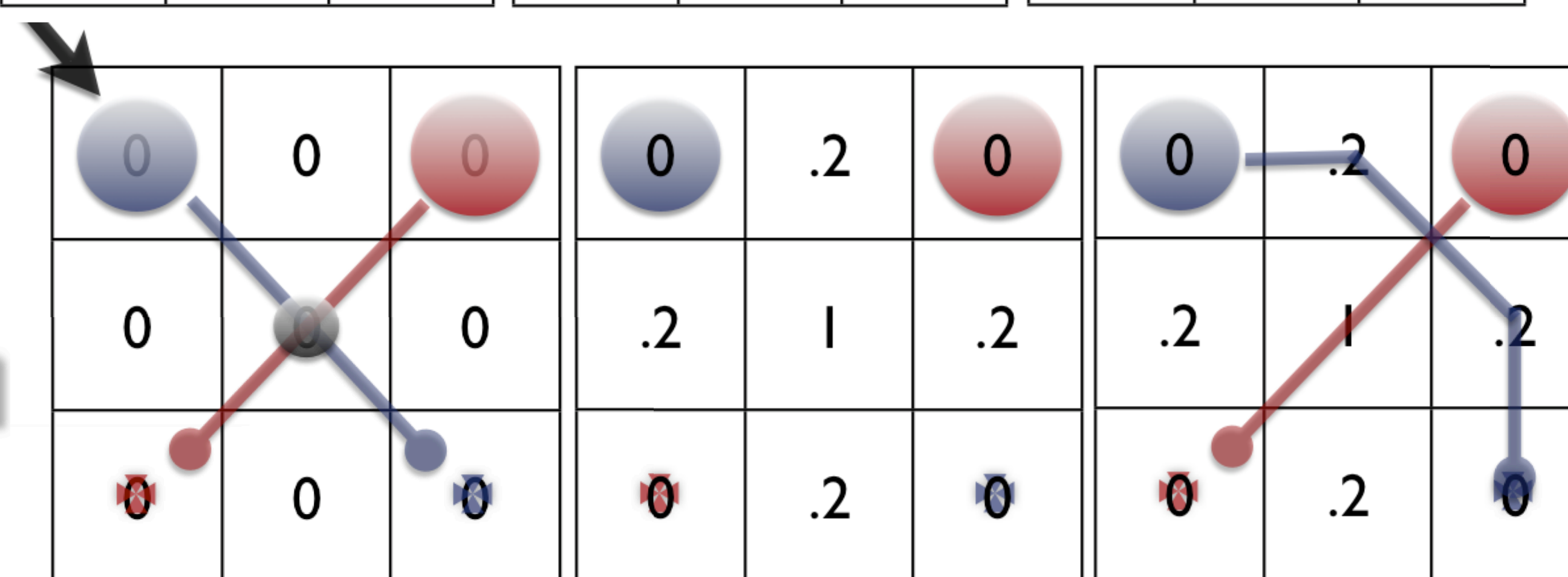
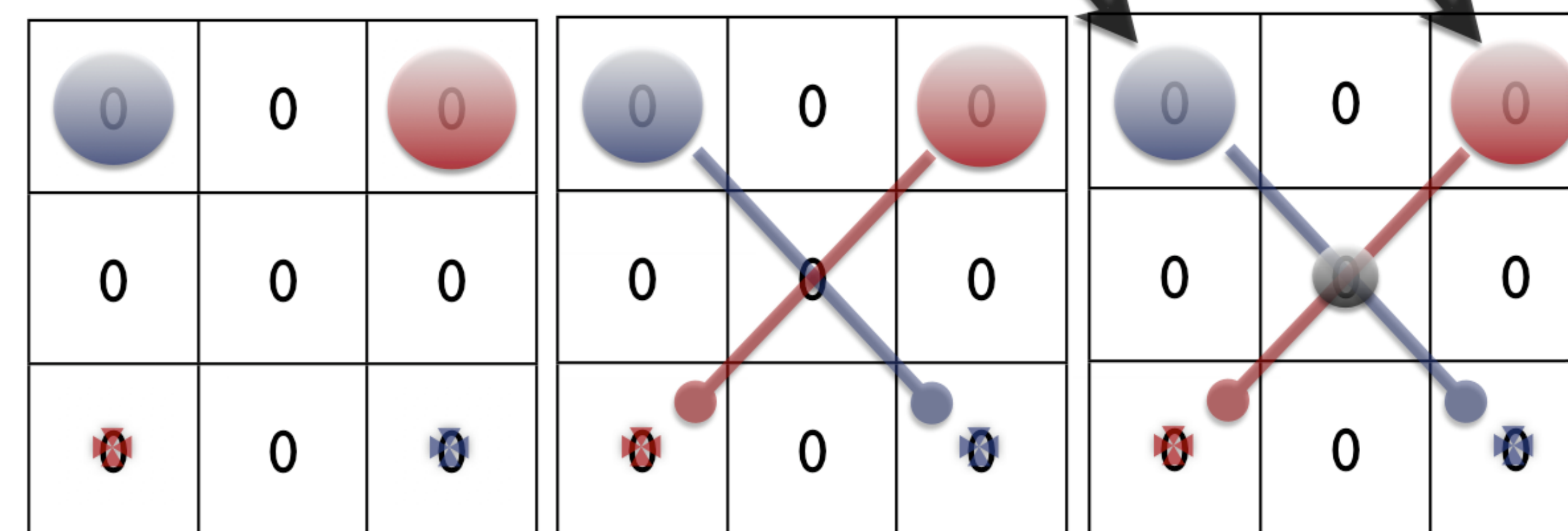
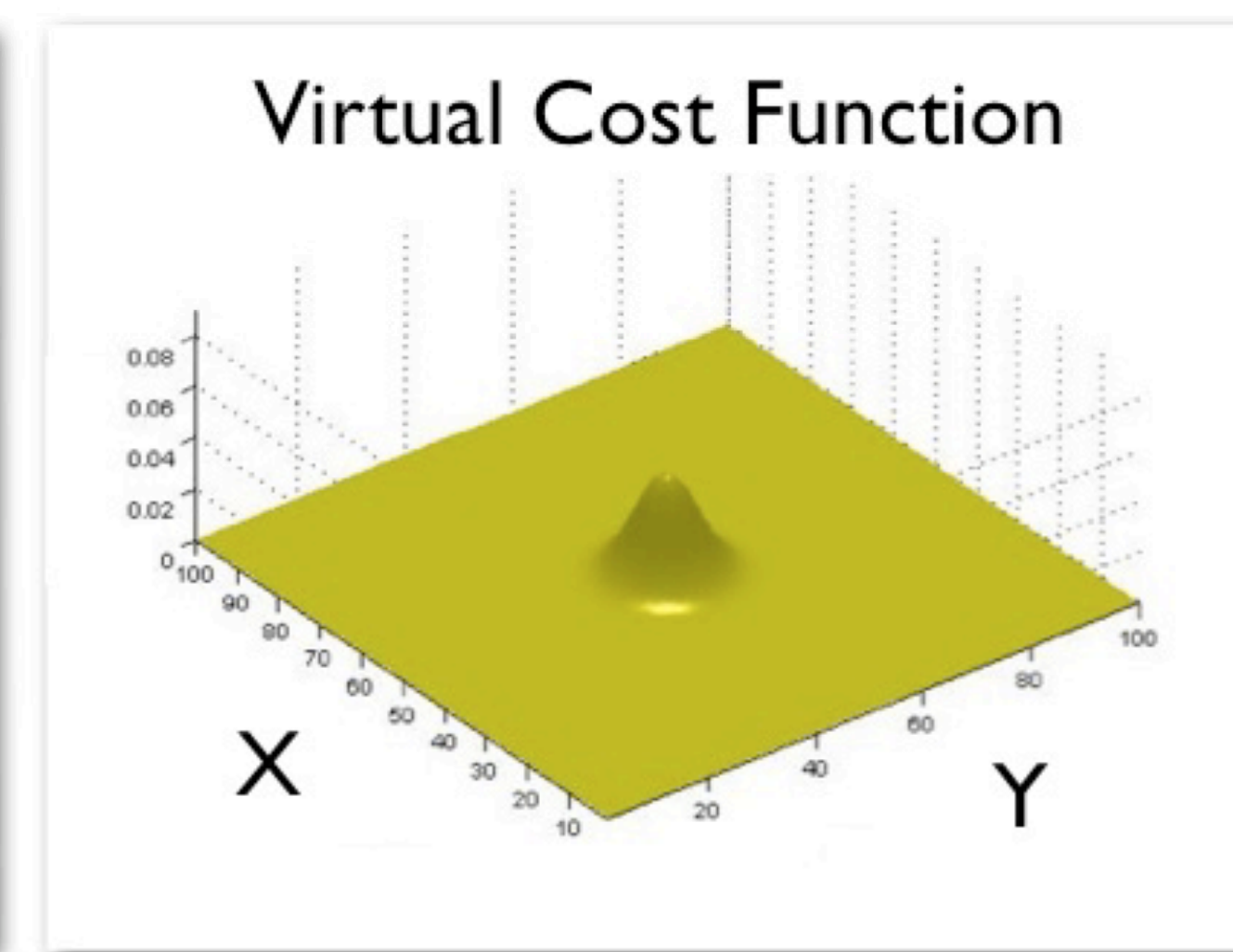
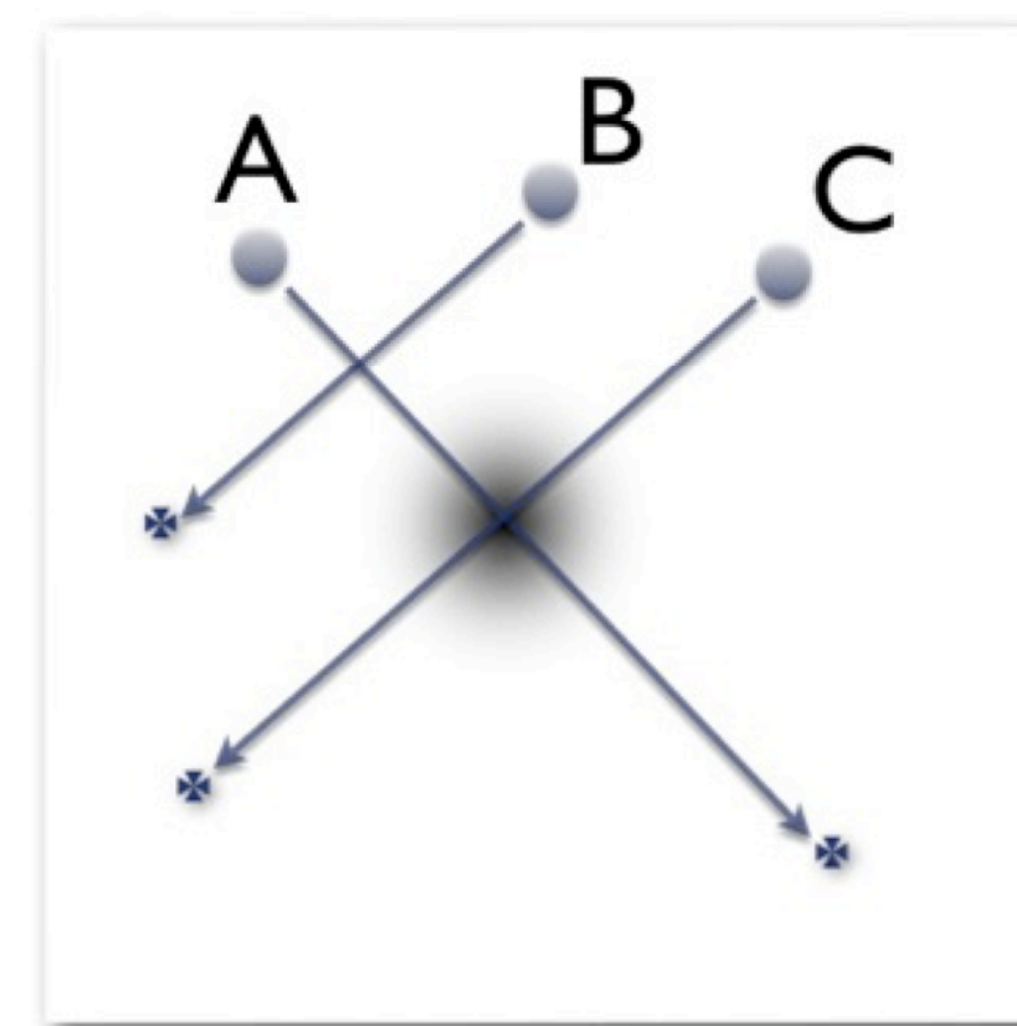
```

BCP
0 Colliding ← True
1 While time is available and Colliding do
2   For each unit i on the map do
3     p ← Limited path from the start to the goal of unit i
4     with maximum length k considering heuristic as h + h'
5     reset the time: t ← 0
6     For each position n on path p do
7       CollisionDetector.add(n, t, i)
8     end for
9   end for
10  C ← CollisionDetector.getCollisions()
11  For each collision c in C do
12    A ← c.units()
13    Delete the unit with highest priority from A
14    For each unit i in A do
15      VirtualHeuristic.add(i, c.x, c.y, c.size)
16    end for
17  end for
18  if C is not empty
19    Colliding ← True
20  else
21    Colliding ← False
22  end while
23  return set of paths with least number of collisions
  
```

```

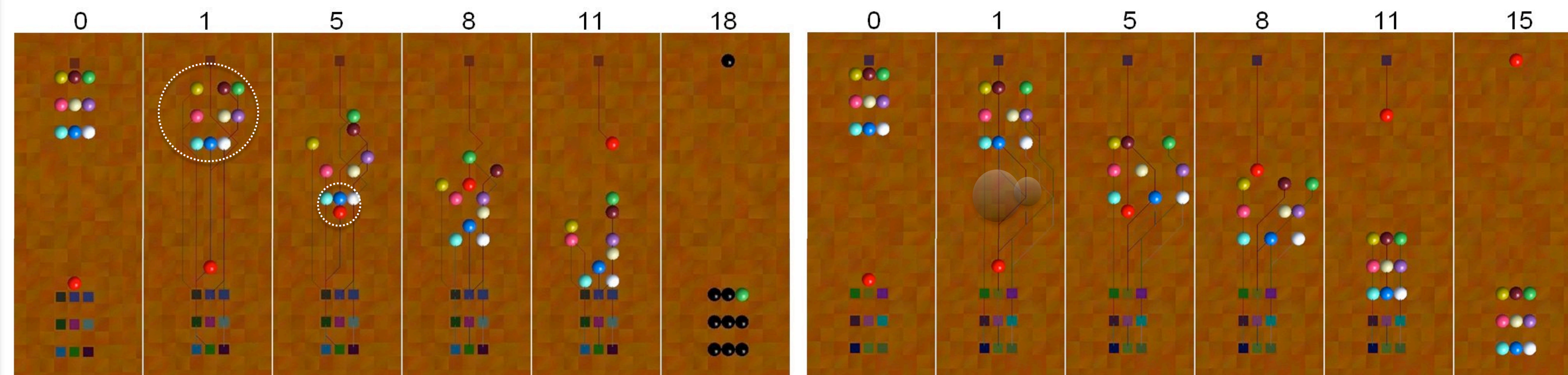
h'(n, i)
1 G ← VirtualHeuristic.Gaussians(i)
2 S ← 0
3 For each function f in G do
4   S ← S + f(n.x, n.y)
5 end for
6 return S
  
```

$$h'' = h + h'$$

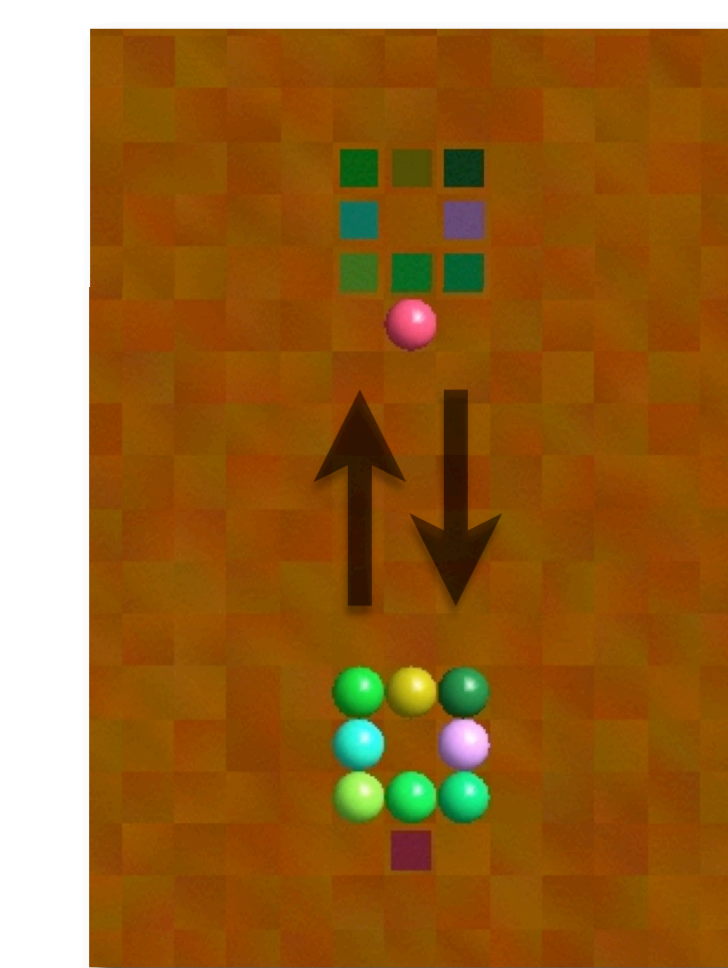
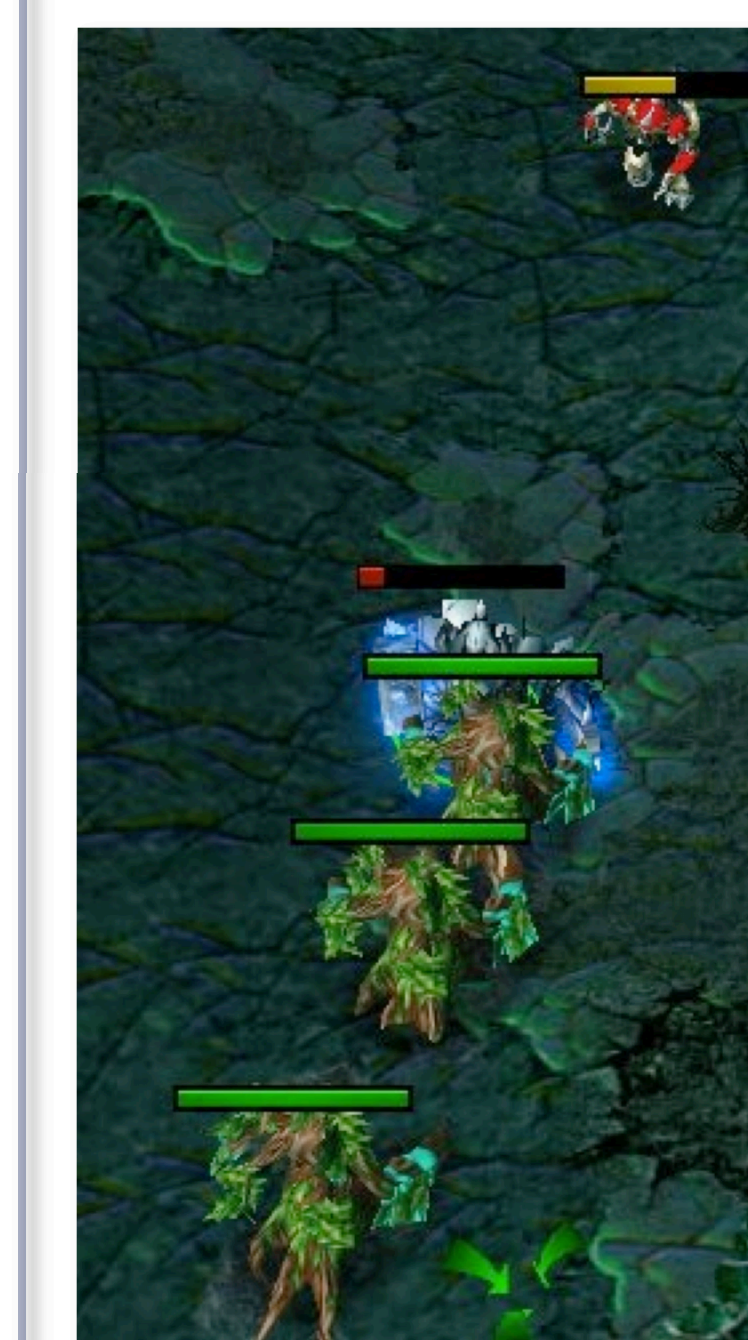


A*

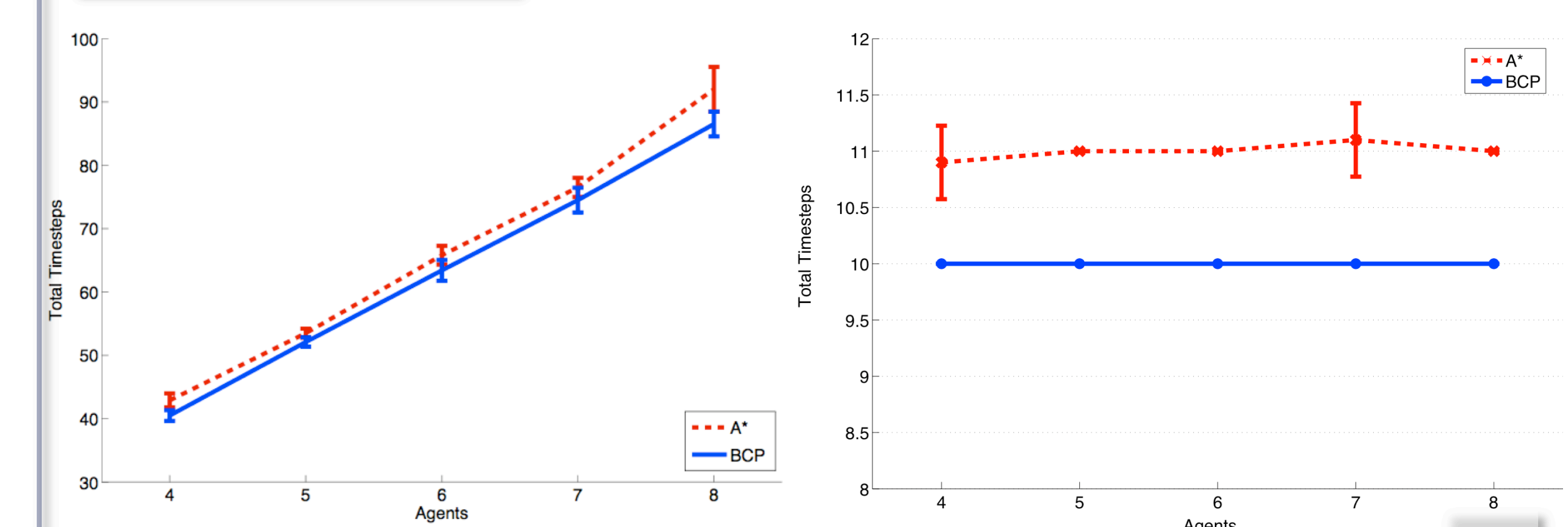
BCP(A*)



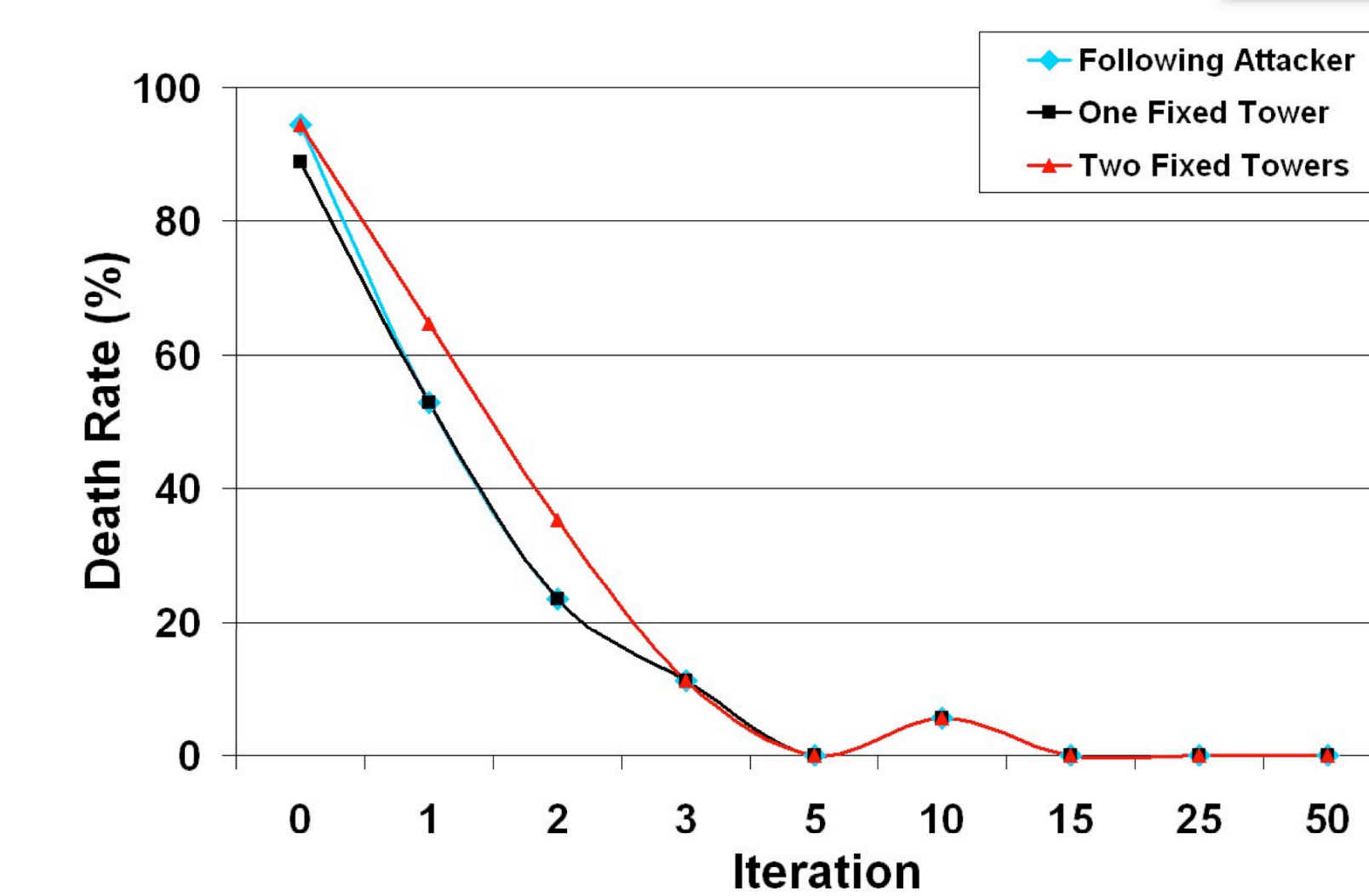
Results



Rescue the Hero



Rescue the Hero Extended



Scaling

