Multi-Agent A*

AAAI 2015 Submission 2327

Abstract

abstract

Notes

For $i = 1, \ldots, N$, agent i has a graph defined by vertex set V_i and edge weights $c_i : V_i \times V_i \to \mathbb{R}^+$. We consider constraints Φ, each defining a relation on the relative visit times of two sets of nodes $\Phi^-, \Phi^+ \subset \cup_i V_i$. There are four constraint types:

- $\min \leq \min$ (opening a door)
- $\max \leq \min$ (closing a door)
- $\max \leq \max$ (restoration)
- $\min \leq \max$ (sequence)

Given individual plans for each agent, an optimal multi-agent plan can be constructed greedily forward in time, when all constraints are of the opening or closing kind. An optimal multi-agent plan can be constructed greedily backward in time when all constraints are of the closing or restoration kind. Since restoration and sequence constraints can be represented in terms of opens and closes, we consider the former as the only types without loss of generality.

Representing restoration contraints in terms of opens and closes: suppose B restores (i.e. cleans after) A. Then create a copy B' of B. Let B' close A, and B' open any agent's goal node. To deal with the boundary condition where neither A nor B are visited, the start node should have a similar copy, reachable at zero cost.

Representing sequence contraints in terms of opens: suppose (A, B) are a mandated sequence. Then create a copy B' of B. Let A open B', and B' open any agent's goal node.

$g_{s_1,\ldots,s_N}(s_i)$ is the minimum time by which agent i can reach state s_i, assuming every agent j will go to s_j along the best known path and wait there forever. Note that the state space for each agent is the Cartesian product of its graph's vertex set, and the set of partial permutations of constraints.

Define $f(s_1, \ldots, s_N) = \max_i g_{s_1,\ldots,s_N}(s_i) + \epsilon h(s_i)$. This key is impractical to compute over all possible tuples. Further approximations will need to be made. Note that $(s_1, \ldots, s_N)$ range over the goal nodes as well as the $OPEN$ list.

Even if we could compute f, does it guarantee $N\epsilon$ optimality? To prove that, it would suffice to show that we only expand nodes whose individual-agent g-values are ϵ-optimal. However, the logic fails because passing through a door-opening node can result in a sudden drastic decrease in the $g_{s_1,\ldots,s_N}$ value, due to constraints being lifted. To remedy this, we could treat doors as being opened after the partial paths to $(s_1, \ldots, s_N)$, at some lower bound time.

Algorithm 1 $MultiAgentA^*()$

```
∀i : s_i := start_i
while g(goal_1, ..., goal_N) > f(s_1, ..., s_N) do
  ∀i : expand(s_i)
  (s_1, ..., s_N) = arg min_{s_1,...,s_N} f(s_1, ..., s_N)
end while
```

Algorithm 2 $expand(s)$

```
for all s' ∈ successors(s) do
  if g(s') > g(s) + c(s, s') then
    g(s') := g(s) + c(s, s')
    bp(s') := s
  end if
end for
```

Algorithm 3 $successors(u, seq_u)$

```
succlist := ∅
for all v such that c(u, v) < ∞ do
  seq_v := seq_u
  if v ∈ Φ as a max constraint then
    append Φ to seq_v
  end if
  if v ∈ Φ as a min constraint and Φ ∉ seq_u then
    append Φ to seq_v
  end if
  push (v, seq_v) onto succlist
end for
return succlist
```