New Valid Inequalities in Branch-and-Cut-and-Price for Multi-Agent Path Finding

Edward Lam, Pierre Le Bodic

BCP
- New algorithm for multi-agent path finding introduced at IJCAI 2019
- BCP is based on branch-and-cut-and-price technique from integer programming
- BCP uses A* for low-level shortest path problem
- Lower bounds provided by linear programming relaxation
- BCP implements two classes of constraints to improve lower bound

Goal
- Only one of:
  - \( a_{\text{goal}} \) reaching its goal location \( l \) at or before time \( t \)
  - \( a_{\text{pass}} \) passing through \( l \) at or after time \( t \)

Exit-entry
- Only one of:
  - \( a_{\text{goal}} \) taking edge \((l_1,t),(l_2,t+1)\)
  - \( a_{\text{pass}} \) taking edge \((l_1,t),(\ast,t+1)\) or \((\ast,t),(l_2,t+1)\) or \((l_2,t),(l_1,t+1)\)

Wait-edge
- Only one of:
  - \( e = (l_1,t),(l_2,t+1) \)
  - \( e' = (l_2,t),(l_1,t+1) \)
  - \( e_{\text{wait}} = (l_1,t),(l_1,t+1) \)

Wait-delay
- Only one of:
  - \( a_1 \) taking edge \((l_1,t),(l_1,t+1)\)
  - \( a_2 \) taking edge \((l_1,t),(\ast,t+1)\) or \((\ast,t),(l_1,t+1)\)

Two-edge
- Only one of:
  - \( a_1 \) taking edge \((l_1,t),(l_2,t+1)\) or \((l_2,t),(l_3,t+1)\)
  - \( a_2 \) taking edge \((l_2,t),(l_1,t+1)\) or \((l_3,t),(l_2,t+1)\)

Contributions
- Five new classes of constraints to improve the lower bound

Experiments
- Graphs showing the performance of BCP compared to other algorithms