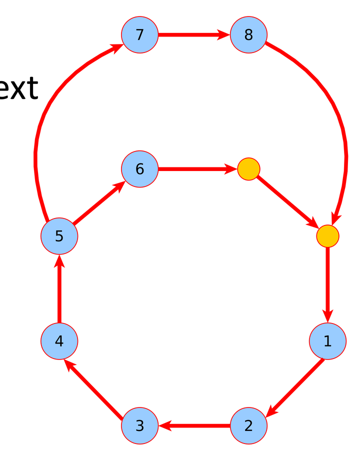
Multi-Agent Path Finding on Strongly Biconnected Digraphs

Adi Botea
IBM Research, Ireland

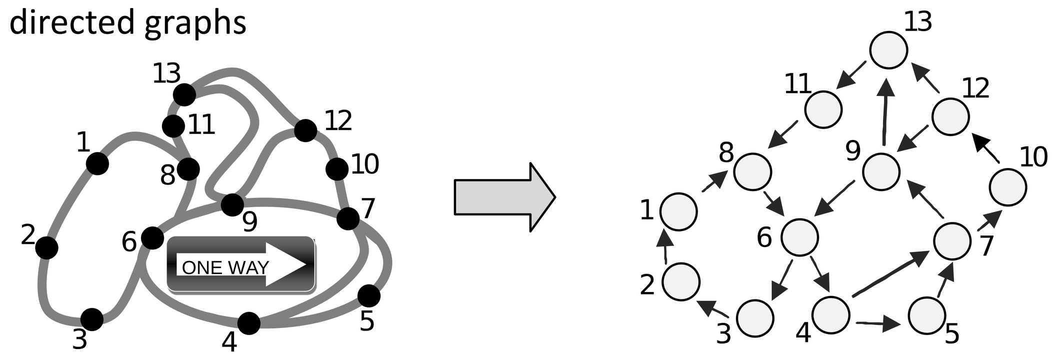
Pavel Surynek
Charles University in Prague, Czech Republic

Introduction

- **A puzzle for you...**
  - An agent (blue) can move to an empty (yellow) next position
- Can you swap agents 1 and 2, but preserve other agents' positions?
- Is it possible to reach any goal configuration?

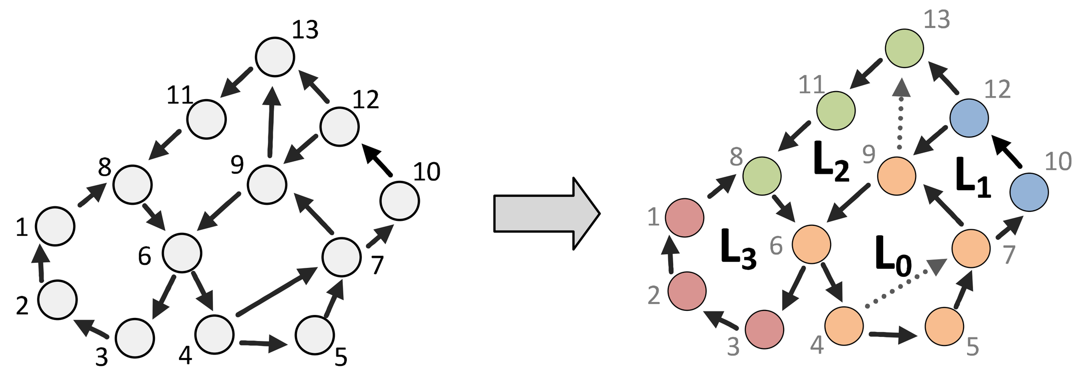


- ***Multi-agent path finding (MAPF) on a digraph***
  - Agents occupy nodes – at most one agent per node at a time
  - Agents can move along directed arcs, if the next node is empty
  - Task: find a sequence of moves from an initial configuration to a goal configuration
- Applications to games, communication in asymmetric networks, robotics, transportation
- Lots of previous work on MAPF on undirected graphs, not so much on directed graphs



Background

- ***Strongly biconnected digraph*** **(Wu and Grumbach 2010) [2] is a directed graph such that:**
  - It is strongly connected, i.e., there is a path x → y and a path y → x for all x,y; and
  - The underlying undirected graph is biconnected, i.e., it is connected and it has no cutting points
- ***Open ear decomposition*** **of a digraph D – a partitioning of D into an ordered sequence of subgraphs [L_0, L_1, … , L_r] such that:**
  - L_0 is a simple cycle ("basic cycle")
  - Each "derived ear" L_i, with i > 0, is a simple path between two distinct nodes from L_0 U … U L_{i-1}, with all L_i's interior nodes and edges disjoint from L_0 U … U L_{i-1}

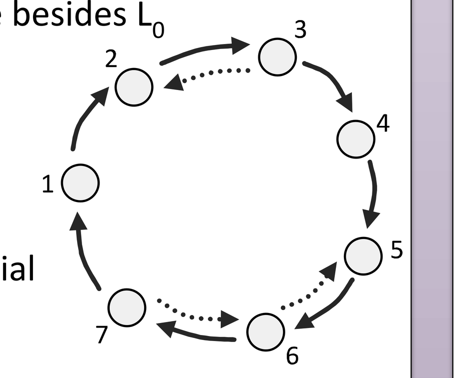


- Theorem (Wu and Grumbach 2010) [2]. A non-trivial digraph D is strongly biconnected iff D has an open ear decomposition.

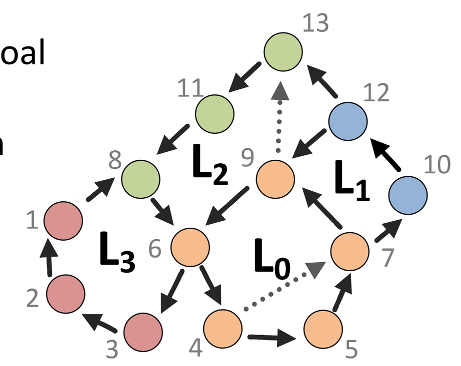
MAPF on Strong Biconnected Digraphs

- **A strongly biconnected digraph either:**
  1) Is a partially bidirectional cycle (see figure); OR
  2) Admits a *regular* open ear decomposition, with L_0 having at least three nodes, and with at least one more node besides L_0

1) MAPF on a partially bidirectional cycle is easy
- Assume one blank is available
- An instance has a solution iff the relative ordering of agents is the same in both the initial and the goal configuration
- Impossible to change ordering of agents in a partially bidirectional cycle



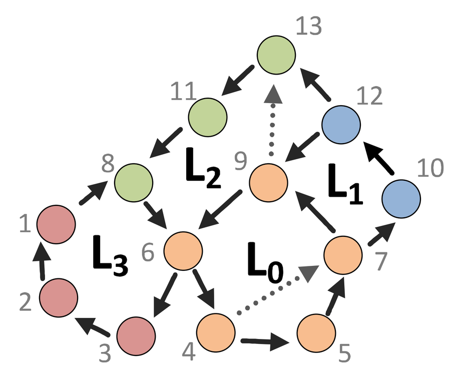
2) MAPF on digraphs with a regular open ear decomposition
- Assume two blanks are available
- Blanks in L_0 both in the initial and the goal configuration – no generality loss
- Let [L_0, L_1, … , L_r] be the decomposition



- **Solving strategy**
  - Solve ears in reverse order
    - L_3, L_2, L_1, L_0 in example
  - Solving a derived ear = reaching goal configuration for its *interior* nodes
  - After solving derived L_i, with i > 1, never again touch its interior → problem gets easier with every ear solved
  - Solving basic cycle L_0 = reaching goal configuration for all its nodes
  - Solving basic cycle L_0 makes use of a derived ear, say L1
  - Must preserve L_1's goal configuration at the end

Solving a Derived Ear

E.g., **solving derived ear L_2**
- Push inside agent a_8, whose goal is node 8
- Push inside agent a_{11}, whose goal is node 11
- Push inside agent a_{13}, whose goal is node 13
- Ear solved!



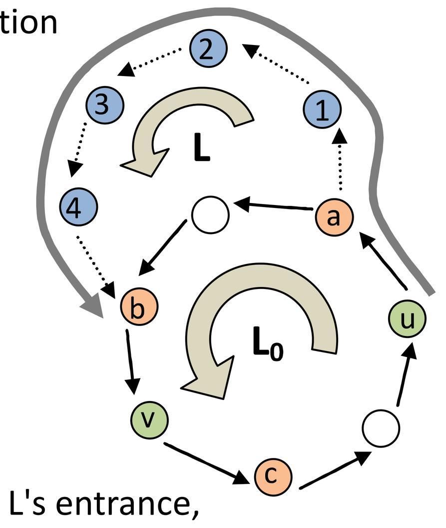
- **How to push an agent inside a derived ear L_i**
  - **Case 1) agent initially outside ear**
    - Put one blank on the first interior position of the ear
    - Bring agent to the entrance of the ear
      - Using only subgraph [L_0, L_1, … , L_{i-1}]
      - Using only one blank
      - Always possible on a strongly biconnected graph (Wu and Grumbach 2010) [2]
    - Push agent inside
  - **Case 2) agent inside ear**
    - Assume we already pushed agent a_8 inside L_2, to node 13
    - Now it's a_{11}'s turn, and a_{11} is located at node 8, inside L_2
    - We need to take a_{11} out, but restore a_8's position
    - Always possible (proof in the paper)
    - And now we reduced case 2 to case 1

diBOX

- **Our algorithm for MAPF on strongly biconnected digraphs**
  - Implementing the strategy presented here
  - Producing suboptimal solutions
    - needs time of O(|V|³) and produces O(|V|³) steps
  - Complete on strongly biconnected digraphs

Solving Basic Cycle

Strategy
- First, re-order agents inside basic cycle
  - Bring agents next to each other, pairwise, as needed
- Then, rotate until reaching goal configuration



- **Bringing two agents u (Mickey) and v (Minnie) next to each other**
  - Assume two blanks are available in L_0
  - Making use of derived ear L
  - Making sure that L's goal configuration is restored at the end
- The steps
  1) Mickey's departure
  - Rotate inside L_0 until Mickey is at the L's entrance, and a blank is at the L's exit

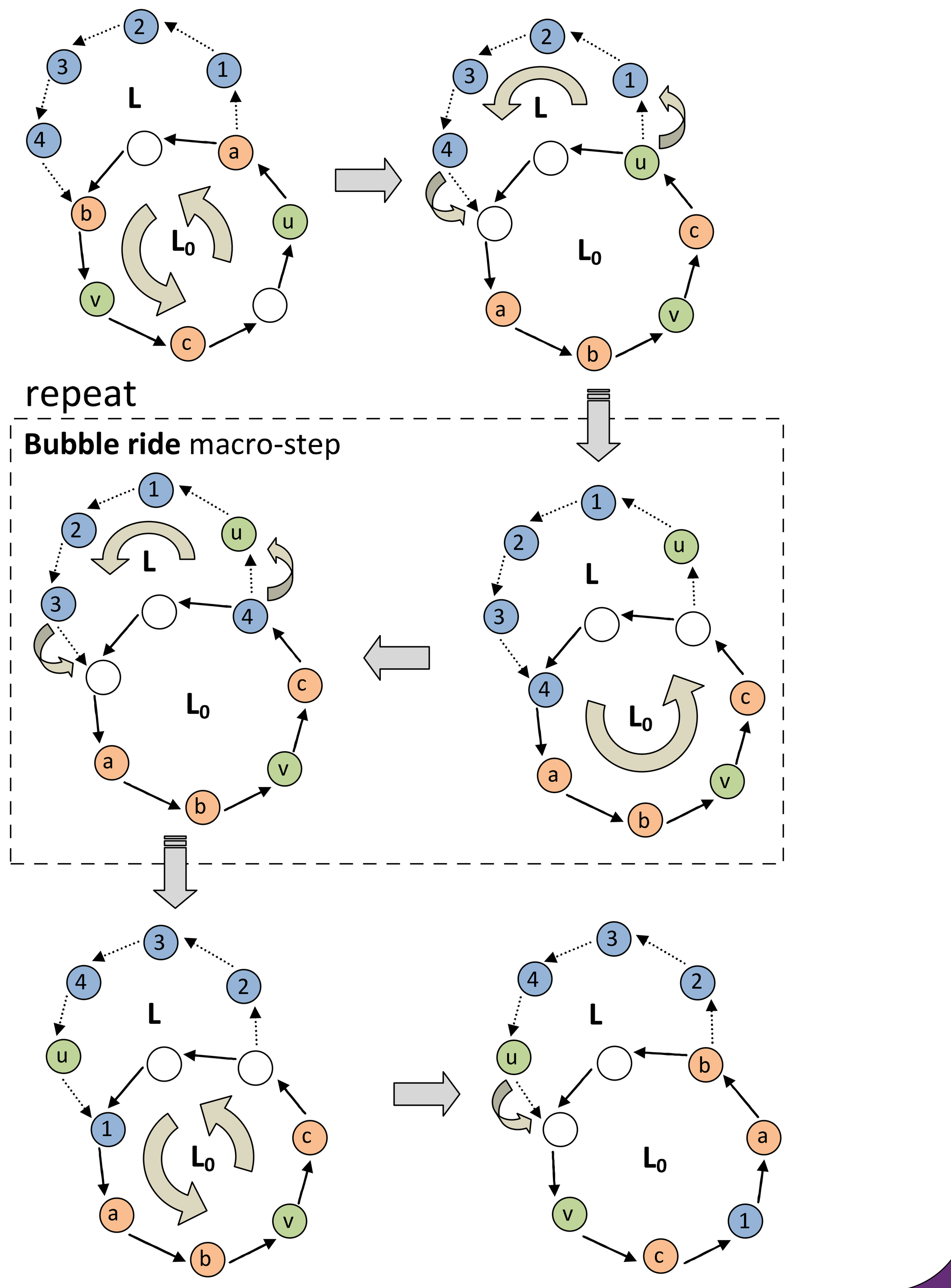
  2) Mickey's admission into the bubble ride
  - Push Mickey inside L

  3) Bubble ride
  - Mickey progresses along L
  - L's configuration is restored step by step
    - Every agent taken out of L is pushed back inside
  - At the end, Mickey is on L's last interior position

  4) Minnie's trip
  - Rotate in L_0 until Minnie is right after the exit from L

  5) Reunification – push Mickey out of L, right after Minnie

  6) Clean up
  - Push inside L its last remaining agent



Conclusion and Future Work

- **Contributions**
  - A new polynomial-time algorithm **diBOX** for MAPF on an interesting class of directed graphs
    - fusion of **BIBOX** [1] algorithm for undirected biconnected graphs
    - and agent relocation procedures on directed graphs [2]
  - A step towards understanding MAPF on directed graphs
- **Open questions**
  - Extension of the **diBOX** algorithm to other classes of directed graphs
  - Performance of **diBOX** on benchmarks and real-life MAPF instances
    - comparison with other algorithms
  - Analysis of quality of solutions generated by **diBOX**
    - how far from the optimum they are
    - approximation algorithms

References
- [1] Surynek, P. : A novel approach to path planning for multiple robots in bi-connected graphs. In IEEE International Conference on Robotics and Automation (ICRA 2009), 2009.
- [2] Wu, Z., and Grumbach, S. 2010. Feasibility of motion planning on acyclic and strongly connected directed graphs. Discrete Applied Mathematics 158(9), 2010.

The 29th AAAI Conference on Artificial Intelligence (**AAAI 2015**), Austin, Texas, USA