Branch-and-Cut-and-Price for Multi-Agent Pickup and Delivery

Edward Lam, Peter Stuckey, Daniel Harabor

ed-lam.com

MONASH University


2178
1348
1348
1375


17
3
8
18
0
6
16
14
12
10
2
1
4
11
19
7
5
9
15
13

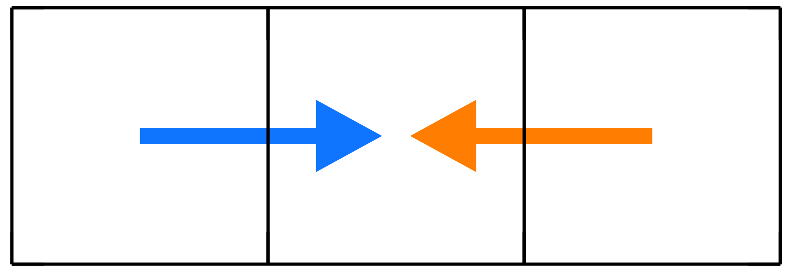

17
3
0
8
18
6
16
14
12
10
2
1
4
11
19
7
5
9
15
13

Multi-Agent Pickup and Delivery

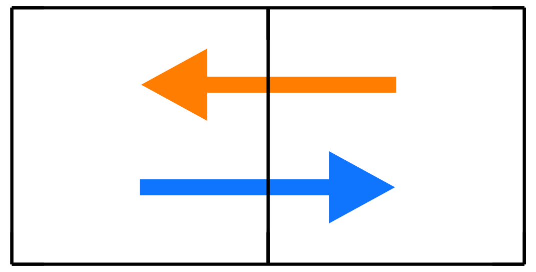
- Contains the Pickup and Delivery Problem with Time Windows (PDPTW)
  - Assign a sequence of pickup-delivery requests to every agent
  - Requests must be visited during time windows
  - Agents have a maximum carrying capacity (capacity 1)

Multi-Agent Pickup and Delivery

- Contains the Multi-Agent Path Finding (MAPF) problem
  - Discrete time
  - Navigate every agent from its start location to its end location
  - Move north, south, east, west or wait
  - Cost 1 for each action (move or wait)
  - Cannot move into obstacles
  - Vertex collision: agents cannot overlap at a location



  - Edge collision: agents cannot cross over to opposite locations



  - Minimize the total number of actions to reach their end locations

Multi-Agent Pickup and Delivery

- Multi-depot: every agent has a different start and end location
- Path-dependent travel time/distance: agents can block a corridor
- Violates first-in first-out property: arriving later can be better

Definitions

- Time horizon T, timesteps $\mathcal{T} = \{1, \dots, T\}$
- Locations $\mathcal{L} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, excludes obstacles
- Agents $\mathcal{A}$
  - Start location L_a^+, end location L_a^- for all $a \in \mathcal{A}$
- Requests $\mathcal{R} = \mathcal{R}^\uparrow \cup \mathcal{R}^\downarrow$, pickup vertices $\mathcal{R}^\uparrow$, delivery vertices $\mathcal{R}^\downarrow$
  - Time window $[\underline{T}_i, \bar{T}_i]$ for all $i \in \mathcal{R}$
  - Location $L_i \in \mathcal{L}$ for all $i \in \mathcal{R}$

Definitions

- Sequencing graph $\mathcal{G}^{\text{seq}} = (\mathcal{V}^{\text{seq}}, \mathcal{E}^{\text{seq}})$
  - Vertices $\mathcal{V}^{\text{seq}} = \{\text{start vertex } \top, \text{end vertex } \bot\} \cup \mathcal{R}$
  - Edges $\mathcal{E}^{\text{seq}}$ between pairs of compatible vertices
- Navigation graph $\mathcal{G}^{\text{nav}} = (\mathcal{V}^{\text{nav}}, E^{\text{nav}})$
  - Vertices $\mathcal{V}^{\text{nav}} = \mathcal{L} \times \mathcal{T}$
  - Edges $\mathcal{E}^{\text{nav}} = \{(((x_1, y_1), t_1), ((x_2, y_2), t_2)) \in \mathcal{V} \times \mathcal{V} : |x_2 - x_1| + |y_2 - y_1| \leq 1 \wedge t_2 = t_1 + 1\}$
  - Every vertex has up to five outgoing edges – north, south, east, west, wait
- Request sequence $s = (\top, r_1, r_2, \ldots, r_n, \bot)$
- Path $p = (L_a^+, l_2, l_3, \ldots, l_{k-1}, L_a^-)$ for some $a \in \mathcal{A}$

Algorithm for Joint Optimization

Intuition

- For every agent, find a sequence of requests and a path navigating the agent to the location of those requests, ignoring collisions
- Assign requests to agents and resolve collisions at the master level

Overview

- Master problem
  - Set partition formulation
  - Select one sequence-path pair for every agent
  - Every request must be completed
  - Every agent must use one path
- Pricing problem
  - Search for a request sequence for every agent (VRP)
  - Search for a path to navigate the agent to each consecutive request (MAPF)
- Separation problems
  - Resolve collisions by adding cuts

Master Problem

Cost of path p

Variable for selection of (s, p)

$$\min \sum_{a \in \mathcal{A}} \sum_{(s,p) \in \Lambda_a} \boxed{c_p} \boxed{\lambda_{(s,p)}} \tag{1}$$

Every request must be completed

$$\boxed{\sum_{a \in \mathcal{A}} \sum_{(s,p) \in \Lambda_a} \alpha_r^s \lambda_{(s,p)} = 1} \quad \forall r \in \mathcal{R} \tag{2}$$

Every agent must use one sequence-path pair

$$\boxed{\sum_{(s,p) \in \Lambda_a} \lambda_{(s,p)} = 1} \quad \forall a \in \mathcal{A} \tag{3}$$

Every sequence-path pair must have integral selection

$$\boxed{\lambda_{(s,p)} \in \mathbb{Z}_+} \quad \forall a \in \mathcal{A}, (s,p) \in \Lambda_a \tag{4}$$

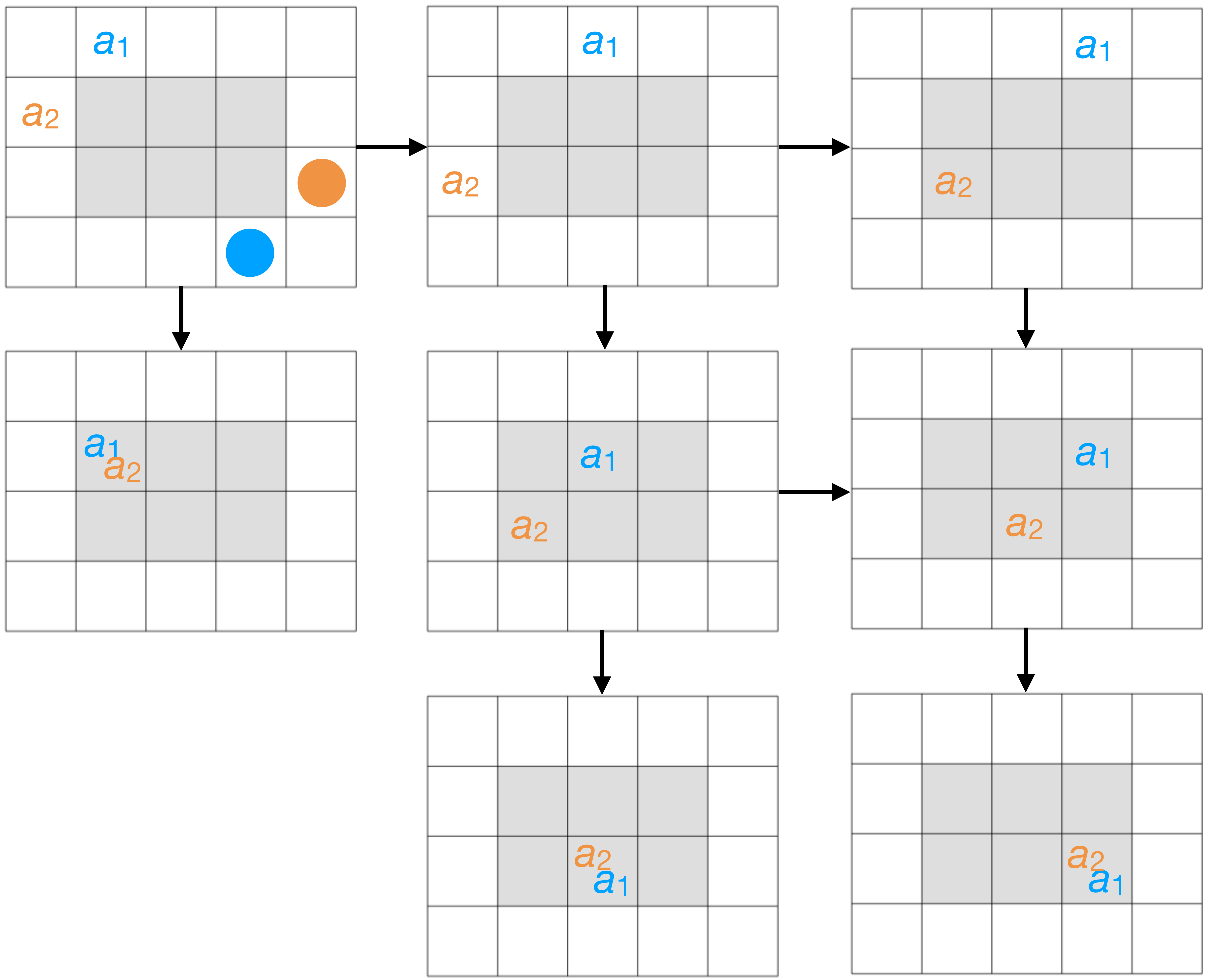
Cuts

- VRP:
  - Subset row
- MAPF (robust)[1]:
  - Vertex
  - Edge
  - Rectangle
  - Corridor
  - Two-edge
  - Exit-entry

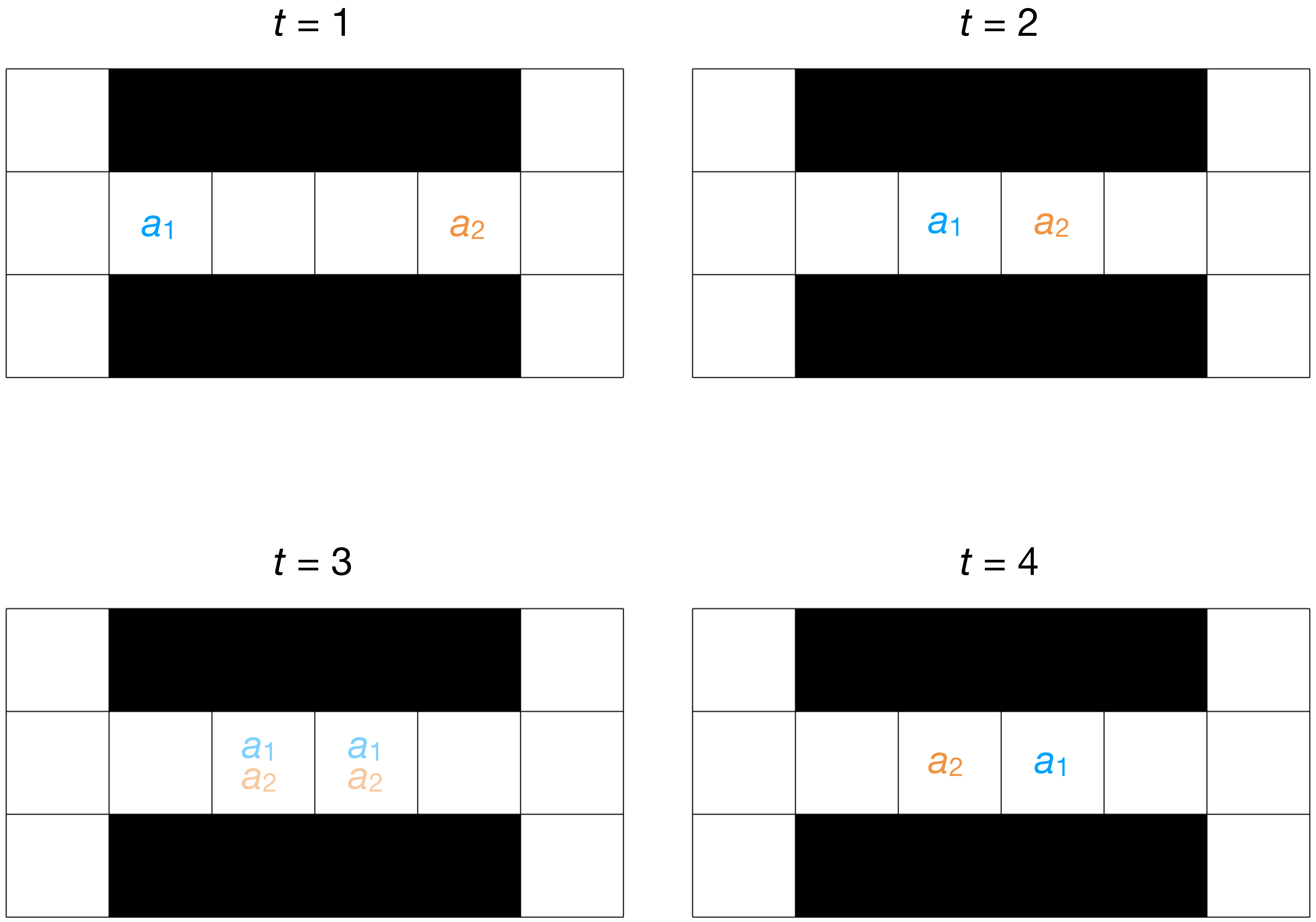
---

[1] E. Lam, P. Le Bodic, D. Harabor, and P. J. Stuckey, “Branch-and-cut-and-price for multi-agent path finding”. Computers & Operations Research, vol. 144, p. 105809, 2022.

Rectangle Cuts



Corridor Cuts



Pricing Problem

- Find a sequence-path pair (s, p) that has negative reduced cost for any agent $a \in \mathcal{A}$
- Solve a two-level shortest path problem
  - High-level:
    - Find a request sequence (path on $\mathcal{G}^{\text{seq}}$)
    - Resource-constrained: reduced cost, time, request (if elementary)
    - Solve using a VRP labeling algorithm
  - Low-level:
    - Navigate the agent from one request to another whenever extending a label
    - Find a path on $\mathcal{G}^{\text{nav}}$
    - No resource constraints
    - Find a Pareto frontier minimizing reduced cost and time
    - Solve using A*

Pricing Problem

- Sequence-path pair (s, p), where $s = (\top, r_1, r_2, \dots, r_n, \bot)$, $p = (L_a^+, l_2, l_3 \dots, L_a^-)$, has reduced cost

High-level

$$\bar{c}_{(s,p)} = c_p - \pi_a - \sum_{i=1}^{n} \rho_{r_i} - \sum_{u \in \mathcal{U}} \sum_{t=1}^{T-1} A^u_{a,((l_t,t),(l_{t+1},t+1))} \sigma_u - \sum_{w \in \mathcal{W}} \left\lfloor \frac{\sum_{i=1}^{n} A^w_{r_i}}{2} \right\rfloor \phi_u$$

Low-level

- $\pi_a, \rho_r, \sigma_u, \phi_w$: dual variable of agent constraints, request constraints, robust MAPF constraints/cuts, subset row cuts
- $\mathcal{U}$: set of robust MAPF constraints/cuts
- $\mathcal{W}$: set of subset row cuts
- $A^u_{a,e}$: coefficient of edge e for agent a in the collision constraint $u \in \mathcal{U}$
- A^w_i: coefficient of vertex i in the subset row cut $w \in \mathcal{W}$

Algorithm for Deferred Optimization

Intuition

- Solve the PDPTW for a request sequence for each agent using their shortest path distances
- Whenever a PDPTW solution is found, check it for MAPF infeasibility or superoptimality
- If so, create a combinatorial Benders cut in the PDPTW master problem

Master Problem

Variable for selection of s

Shortest path cost for sequence s

Variable for VRP and MAPF cost difference

$$\min \sum_{a \in \mathcal{A}} \sum_{s \in \Lambda_a} c_s \lambda_s + \theta \quad (5)$$

Every request must be completed

$$\sum_{a \in \mathcal{A}} \sum_{s \in \Lambda_a} \alpha_r^s \lambda_s = 1 \quad \forall r \in \mathcal{R} \quad (6)$$

Every agent must use one sequence

$$\sum_{s \in \Lambda_a} \lambda_s = 1 \quad \forall a \in \mathcal{A} \quad (7)$$

Every sequence must have integral selection

$$\lambda_s \in \mathbb{Z}_+ \quad \forall a \in \mathcal{A}, s \in \Lambda_a \quad (8)$$

Non-negative cost difference

$$\theta \geq 0 \quad (9)$$

Benders Problem

1. Get the used arcs of a VRP feasible solution

$$\mathcal{F} = \left\{ (a, i, j) \in \mathcal{A} \times \mathcal{E}^{\text{seq}} : \sum_{s \in \Lambda_a} \alpha_{i,j}^{s} \lambda_s = 1 \right\}$$

2. Enforce these sequences in the MAPF problem and solve
3. If $\mathcal{F}$ is infeasible, adds the feasibility cut

$$\sum_{(a,i,j) \in \mathcal{F}} \sum_{s \in \Lambda_a} \alpha_{i,j}^{s} \lambda_s \leq |\mathcal{F}| - 1$$

4. If $\mathcal{F}$ is superoptimal, adds the optimality cut

$$\sum_{(a,i,j) \in \mathcal{F}} \sum_{s \in \Lambda_a} \alpha_{i,j}^{s} \lambda_s - \frac{1}{\delta} \theta \leq |\mathcal{F}| - 1$$

   where δ is the cost difference

Two-Stage Heuristic

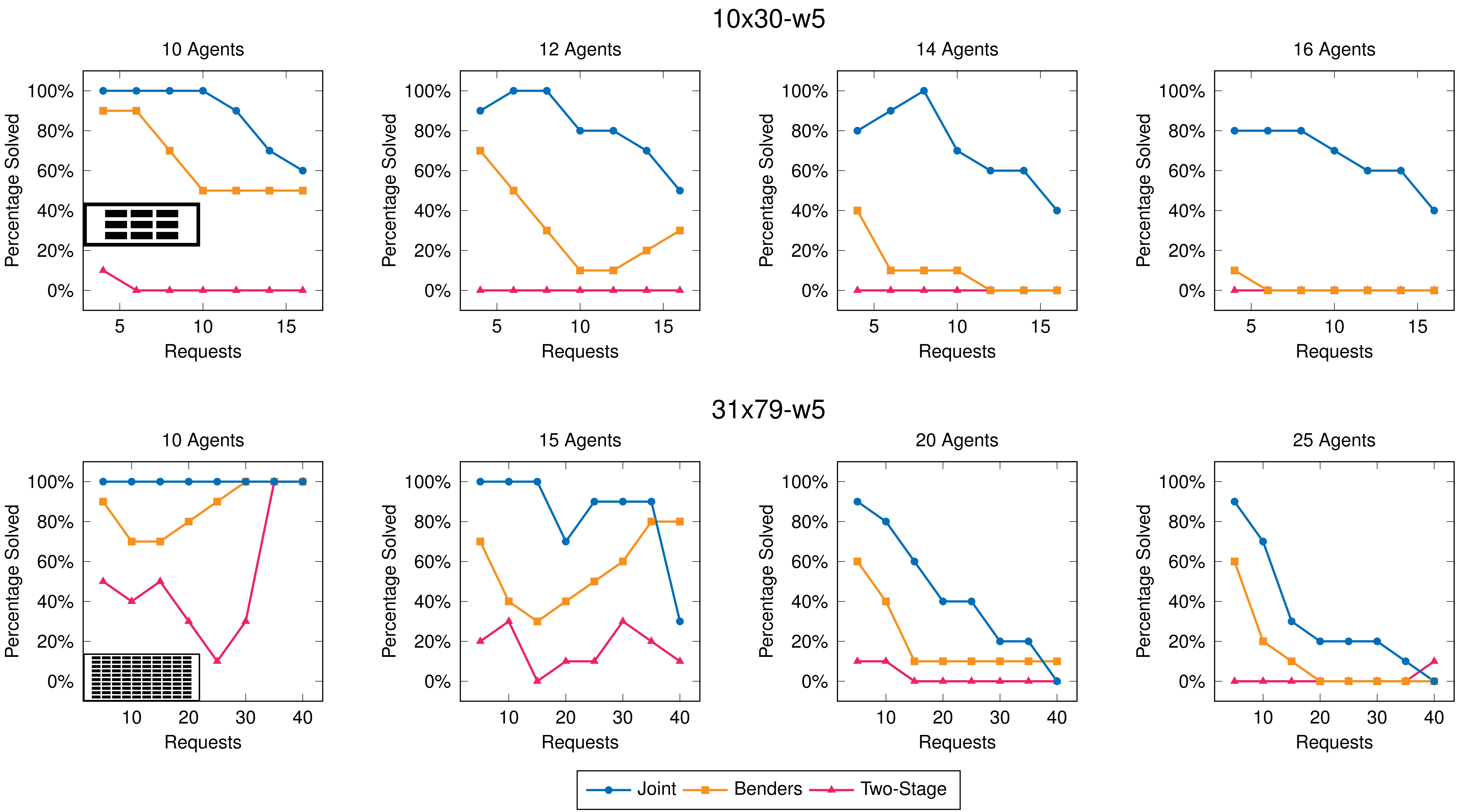
Two-Stage Heuristic

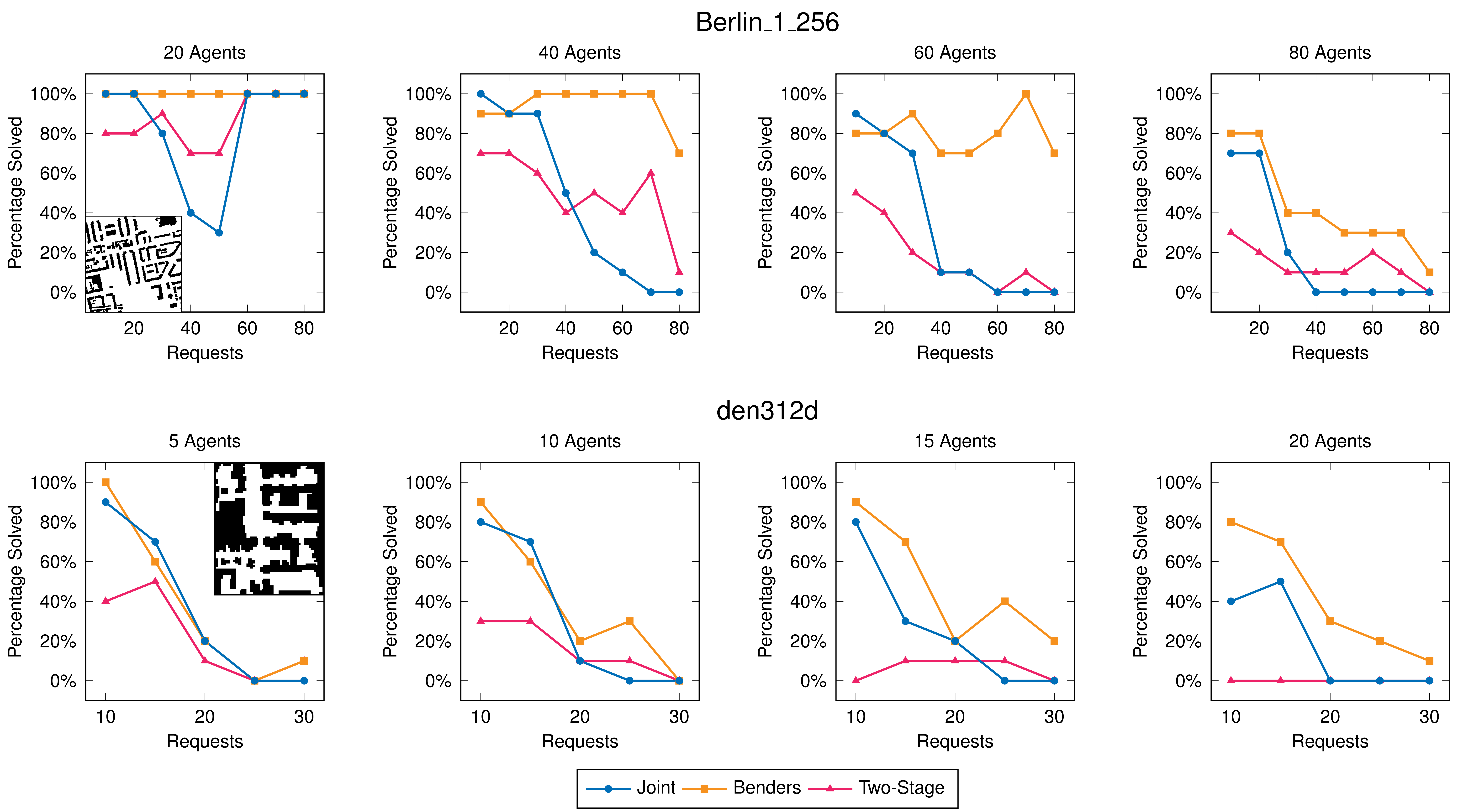
- Solve the PDPTW optimally
- Solve the MAPF using the request sequences from the PDPTW
- Compute a gap using the PDPTW lower bound and MAPF upper bound

Experimental Results

Set-Up

- AMD EPYC Rome 32-core 2.25 GHz
- 128 GB memory
- Gurobi 10.1 LP solver, SCIP 8.0.3 MIP solver
- 1 hour CPU time
- 1120 instances across 4 maps


10x30-w5
10 Agents
12 Agents
14 Agents
16 Agents
31x79-w5
10 Agents
15 Agents
20 Agents
25 Agents
Percentage Solved
Requests
0%
20%
40%
60%
80%
100%
5
10
15
20
30
40
Joint
Benders
Two-Stage


Berlin_1_256
20 Agents
40 Agents
60 Agents
80 Agents
Percentage Solved
Requests
den312d
5 Agents
10 Agents
15 Agents
20 Agents
Joint
Benders
Two-Stage

Future Work

- Cuts over the VRP and MAPF intersection polyhedron
- Better explanation of infeasibility and superoptimality for Benders method (irreducible infeasible subsystem, conflict analysis?)

Branch-and-Cut-and-Price for Multi-Agent Pickup and Delivery

Edward Lam, Peter Stuckey, Daniel Harabor

ed-lam.com

MONASH University