Intractability of Optimal Multi-Agent Pathfinding on Directed Graphs

Xing Tan[1] **and Pascal Bercher**[2]

[1]Dept. of CS, Lakehead University, Canada; [2]School of Computing, The Australian National University

xing.tan@lakeheadu.ca; pascal.bercher@anu.edu.au

1. A Directed Multi-Agent Path-Finding (diMAPF) Example

- Initially at $\mathcal{I} \equiv \mathcal{S}_0$, the green agent A_g is at vertex a, and the blue agent A_b is at b. The goal for A_g is c, for A_b is d. In between $\mathcal{I}$ and $\mathcal{G}$, A_g waits at vertex a for A_b to pass vertex e first.

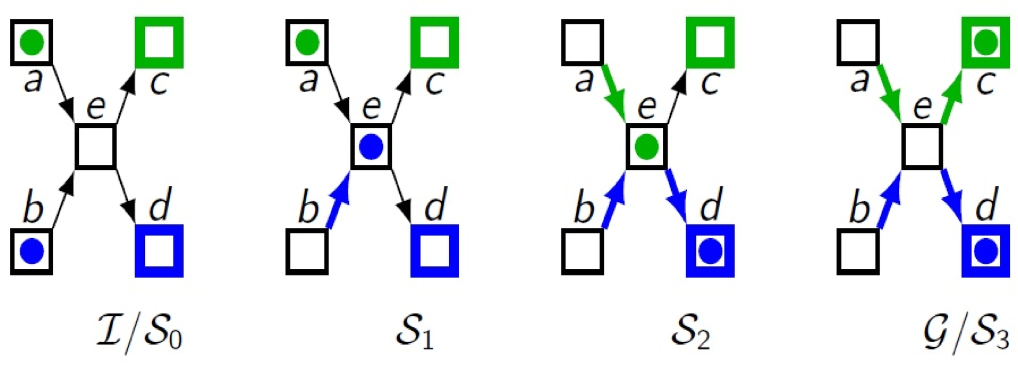

Figure 1: *A diMAPF example. Initially at $\mathcal{I} \equiv \mathcal{S}_0$, the green agent A_g is at vertex a, and the blue agent A_b is at b. The goal for A_g is c, for A_b is d. In between $\mathcal{I}$ and $\mathcal{G}$, A_g waits at vertex a for A_b to pass vertex e first.*

2. Definitions

- Let $\mathcal{A}$ be a finite set of agents, and $G = \langle V, E\rangle$ a directed graph, where V is a finite set of vertices and $E \subseteq V \times V$ is a finite set of directed edges.
- An agent in $\mathcal{A}$ can move from $v_i \in V$ to $v_j \in V$ if $(v_i, v_j) \in E$ is an edge in the directed graph G.
- A state $\mathcal{S}$ defines a distribution of all agents from $\mathcal{A}$, in vertices from V.
- Time is measured in steps. A step σ defines a step-wise movement of all agents, which changes a state $\mathcal{S}$ into its successor $\mathcal{S}^{succ}$.
- It is required that the movement of all agents in σ, between $\mathcal{S}$ and $\mathcal{S}^{succ}$, should be applicable ones, and the applicability of agent movements is defined by the principles of precondition and frame axioms (in classical AI planning)

MAPF A four-tuple $\langle G, \mathcal{A}, \mathcal{I}, \mathcal{G}\rangle$, where G is an undirected graph, $\mathcal{A}$ is a set of agents, $\mathcal{I}$ is the initial situation, and $\mathcal{G}$ is the goal situation. Is there a sequence $\vec{\mathcal{S}}$ that moves agents in $\mathcal{A}$ from $\mathcal{I}$ to $\mathcal{G}$?

diMAPF$^{\mathcal{R}}$ A directed MAPF problem on a set of restrictions $\mathcal{R}$, e.g.,:

d≤3 Let "$dgr(i, j)$" denote a vertex in a digraph having in-degree i and out-degree j. For any vertex v in the graph, if $dgr(i, j)$ holds for v, then $i + j \leq 3$ and $i, j \leq 2$ hold as well.

dag G is acyclic, i.e., a DAG.

pl G is planar. That is, it can be drawn on the plane in a way such that none of its edges intersect with each other.

sc G is strongly connected, i.e., every pair of vertices u and v should have a path in each direction between them.

uc G is unilaterally connected, i.e., every pair of vertices u and v in G should have a path in at least one direction between them.

wc G is weakly connected, i.e., if there is a path between every pair of vertices u and v in the underlying undirected graph.

3. Intractability of Decisional diMAPF Problems

Theorem 1 *The problem diMAPF is NP-hard, and in PSPACE; NP-complete, if $\mathcal{G}$ is a directed acyclic graph (**dag**); NP-complete, if $\mathcal{G}$ is a strongly connected digraph (**sc**). (Thm 1, Props 2/3, and Thm 4 in (Nebel 2020[1]), Thms 18/19 in (Nebel 2023[2]))*

Theorem 2 *A Rectilinear-Planar Monotone Sided 3SAT (RPMS-3SAT) instance is a Boolean formula in the 3SAT format, rectilinearly-planar, and monotone. In addition, the clauses are sided: All positive/negative clauses are on the side above/below the variable line, respectively. RPMS-3SAT is NP-complete. (Thm 1 in Mark de Berg and et. al. 2010[3])*

Theorem 3 *diMAPF^{dag,pl} is NP-complete.*

Corollary 4 *diMAPF$^{dag,pl,\mathbf{d}\leq 3}$ is NP-complete.*

Definition 1 *A Rectilinear-Planar Monotone Sided 3SAT (RPMS-3SAT) instance is a Boolean formula in the 3SAT format, rectilinearly-planar, and monotone. In addition, the clauses are sided: All positive/negative clauses are on the side above/below the variable line, respectively.*

An example *RPMS-3SAT* instance, where the clause nodes are in grey color, and variable nodes are the white ones. The formula is

$$(x_1 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_4)$$

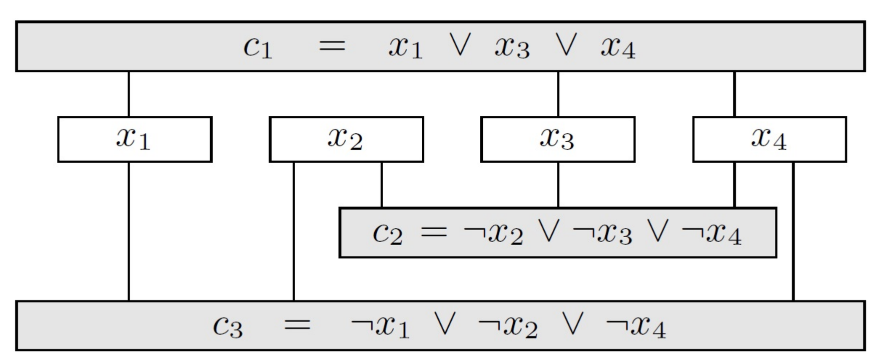

Figure 2: *An example RPMS-3SAT instance.*

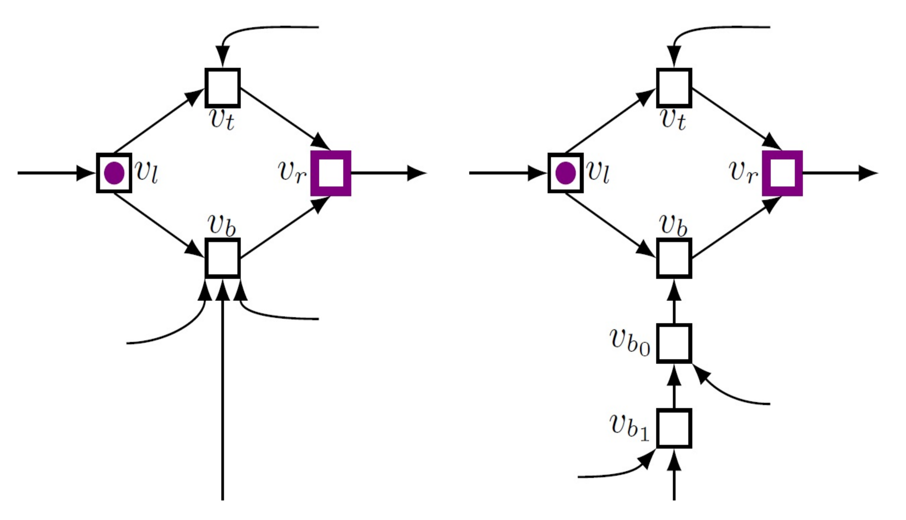

Figure 3: *A diamond component, constructed from a variable in the original RPMS-3SAT instance. For each variable in RPMS-3SAT, one and only one diamond component is generated. All these diamonds are to be connected into a chain (as shown in Figure 4). A diamond variant to the one in (a). Two vertices (v_{b_0} and v_{b_1}) are added below v_b, and the variant now satisfies "d≤3", a constraint that is studied in Theorem 8.*

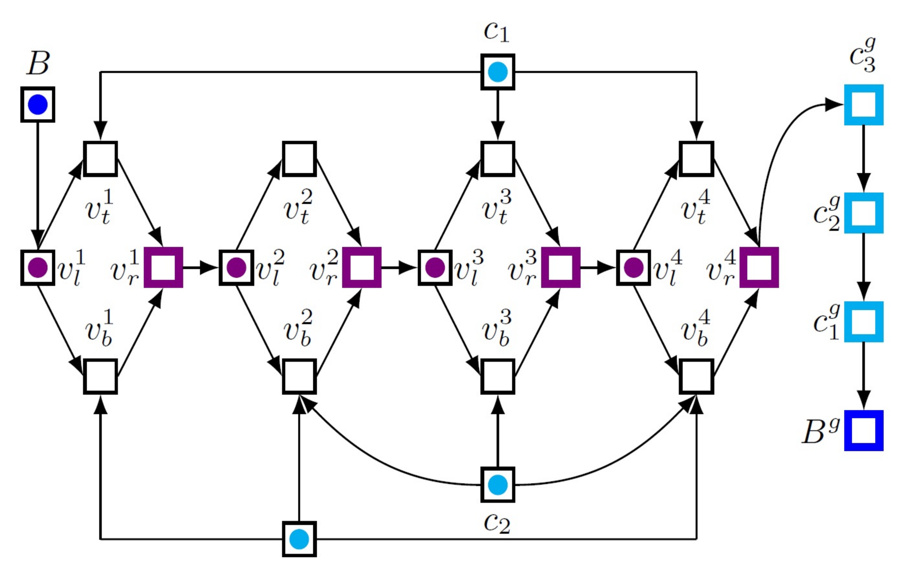

Figure 4: *The resulting diMAPF$^{dag.pl}$ instance $\langle G_{(dag,pl)}, \mathcal{A}, \mathcal{I}, \mathcal{G}\rangle$, converted from the RPMS-3SAT instance in Figure 2. Note in particular the graph $G_{(dag,pl)}$ is a DAG, and is planar.*

4. Optimal diMAPF (OdP)

- Initially, A_g is at a, and A_b is at b. The goal for A_g is c, for A_b is d.
- Path lengths for both agents are 2: $|p_g| = 2$ and $|p_b| = 2$, and $\mathcal{P} = \{p_g, p_b\}$, containing all paths for all agents.
- The maximal distance of the example $\underset{p \in \mathcal{P}}{\operatorname{argmax}}(|p|)$, is 2.
- Total distance is $2 + 2 = 4$.
- Total number of steps from $\mathcal{I}$ to $\mathcal{G}$ (i.e., makespan) is $|\Sigma| = 3$.
- The minimum **max**imal distance of the example is actually 2.
- The minimum **tot**al distance is $2 + 2 = 4$.
- The minimum **makes**pan is actually 3.

Definition 2 (*diMAPF*$^{\mathcal{R}}_{\langle O, \mathbf{B}\rangle}$) *diMAPF$^{\mathcal{R}}_{\langle O,B\rangle}$ is a diMAPF problem subject to restrictions $\mathcal{R}$ and an optimization criterion of type O, which is bounded by $B \in \mathbb{Z}^+$.*
To decide diMAPF$^{\mathcal{R}}_{\langle O,B\rangle}$ with $O \in \{ms, max, tot\}$ means to answer: Is there a $\vec{\mathcal{S}}$, such that

- *If $O =$ ms, then: $|\Sigma| \leq B$?*
- *If $O =$ max, then: $\underset{p \in \mathcal{P}}{\operatorname{argmax}}(|p| \leq B)$?*
- *If $O =$ tot, then: $\left(\sum_{p \in \mathcal{P}} |p|\right) \leq B$?*

5. Complexity of OdP With Makespan Restrictions

Definition 3 *Given a diMAPF$^{\mathcal{R}}_{\langle ms, lb_{ms}\rangle}$ problem, lb_{ms} is the lower bound makespan, if there does not exist a $\vec{\mathcal{S}}$, which is a solution to diMAPF$^{\mathcal{R}}$, and $|\Sigma| < lb_{ms}$.*

Definition 4 (*LB$_{ms}$* of *diMAPF*$^{\mathcal{R}}$) *Given a diMAPF$^{\mathcal{R}}$ problem, and a set $\mathcal{P}_{min}$ containing the shortest paths from their initial vertices to their goal vertices, for all the agents in the problem. Suppose $\hat{p} \in \mathcal{P}_{min}$ is a path whose length is the maximal one among all paths, LB$_{ms} = |\hat{p}|$.*

Proposition 1 *Given a diMAPF$^{\mathcal{R}}$, $\mathcal{P}_{min}$, thus LB$_{ms}$, can be calculated in poly-time (Dijkstra's Algorithm to find shortest paths).*

Theorem 5 *diMAPF$^{dag,pl}_{\langle ms, LB_{ms}\rangle}$ is NP-complete.*

6. OdP with Restrictions on Path Length

Definition 5 *We first identify respectively in the following two lower bounds for maximal individual travel distance, and for total distance.*

- ***(lower bound of maximal distance: LB$_{max}$)*** *We can again find out $\mathcal{P}_{min}$. Among them, we pick up $\hat{p}$ the longest one, whose length $|\hat{p}| =$ LB$_{max}$, which serves as the lower bound for B (justification is similar to the one for lb_{ms}/LB$_{ms}$).*
- ***(lower bound of total distance: LB$_{tot}$)*** *The actual LB$_{tot}$ should be instantiated into LB$_{tot} = \sum_{p \in \mathcal{P}_{min}} |p|$. Any path $p \in \mathcal{P}_{min}$ is the shortest one for its agent, the sum of all path lengths in any solution to the given problem thus can not be shorter than LB$_{tot}$.*

Theorem 6 *diMAPF$^{dag,pl}_{\langle max, LB_{max}\rangle}$ is NP-complete.*

Theorem 7 *diMAPF$^{dag,pl}_{\langle tot, LB_{tot}\rangle}$ is NP-complete.*

Theorem 8 *All three of the following problems are NP-complete: diMAPF$^{dag,pl,\mathbf{d}\leq 3}_{\langle ms, LB_{ms}\rangle}$, diMAPF$^{dag,pl,\mathbf{d}\leq 3}_{\langle max, LB_{max}\rangle}$ and diMAPF$^{dag,pl,\mathbf{d}\leq 3}_{\langle tot, LB_{tot}\rangle}$.*

7. Conclusions

This research investigates the complexity of decisional/optimal *diMAPF* problems.

- the *diMAPF* problems are subject to the optimization criterion of *ms*, *max*, and *tot*; Meanwhile,
- a variety of different special cases of the problem are considered. For example, the digraphs are restricted to be planar, or/and are with bounded vertex-degrees.
- Surprisingly, even if it is applied with the smallest-possible value, optimal *diMAPF* remains to be NP-hard, for the criteria of *ms*, *max*, and *tot*, respectively.
- These complexity results have important real-world implications. For instance, we now know that when on a two-dimensional space only (**pl**), with one-way roads only (**dag**), and at most three-branch junctions only (**d≤3**), MAPF is already computationally challenging.

8. Acknowledgements

We would like to thank the referees for dedicating their time to read our paper and for providing us with valuable feedback. Their constructive input has contributed to the improvement of this paper.

[1]Bernhard Nebel, "On the computational complexity of multi-agent pathfinding on directed graphs", in Proc. of the 30th ICAPS, pp. 212–216, (2020).
[2]Bernhard Nebel, "The small solution hypothesis for MAPF on strongly connected directed graphs is true", in Proc. of the 33rd ICAPS, (2023).
[3]Mark de Berg and Amirali Khosravi, "Optimal binary space partitions in the plane", in Computing and Combinatorics, pp. 216–225, Berlin, Heidelberg, (2010).