SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Perception Field Based Imitation Learning for Unlabeled Multi-Agent Path Finding

Wenjie Chu[1,2], Ailun Yu[1,2], Wei Zhang[1,2*], Haiyan Zhao[1,2*] & Zhi Jin[1,2*]

[1]*School of Computer Science, Peking University, Beijing* 100871*, China;*
[2]*Key Lab. of High-Confidence Software Technologies (Peking University), Ministry of Education, Beijing, China*

Appendix A Optimization Objectives and Equivalence Proof

As mentioned in the main text, three optimization objectives about makespan, flowtime, and sum-of-costs in the offline scenario are actually the degenerated forms of the three optimization objectives about average throughput, average delay time, and average travel distance in the online scenario, respectively. Here, we give three brief proofs of the equivalence between online and offline optimization objectives.

Proposition 1. Maximizing average throughput in online scenarios is equivalent to minimizing makespan in offline scenarios.

Proof. As defined in the main text,

$$throughput(\Gamma) \doteq (1/\mathrm{T}) \sum_{\mathrm{t}=1}^{\mathrm{T}} |\mathcal{G}_{\mathrm{t}}^{\mathrm{b}-}| \tag{A1}$$

$$makespan(\Gamma) \doteq T \tag{A2}$$

In offline scenarios, agents reach and complete all targets at the end, i.e., $\sum_{t=1}^{T} |\mathcal{G}_t^{b-}| = |\mathcal{G}_0^e|$. Since $\mathcal{G}_0^e$ is a static set initialized at the beginning of an offline task and $|\mathcal{G}_0^e|$ is a static value, maximizing $throughput(\Gamma) = \frac{|\mathcal{G}_0^e|}{T}$ is obvious equivalent to minimizing $makespan(\Gamma) = T$.

Proposition 2. Minimizing average delay time in online scenarios is equivalent to minimizing flowtime in offline scenarios.

Proof. As defined in the main text,

$$delay(\Gamma) \doteq \frac{1}{|\mathcal{G}_0^e| + \sum_{t=1}^{T-1} |\mathcal{G}_t^{e+}|} \sum_{t=1}^{T} |\mathcal{G}_t^b| \tag{A3}$$

$$flowtime(\Gamma) \doteq \sum_{t=1}^{T} |\mathcal{G}_t^b| \tag{A4}$$

In offline scenarios, no targets are added after time step 0, i.e., $|\mathcal{G}_0^e| + \sum_{t=1}^{T-1} |\mathcal{G}_t^{e+}| = |\mathcal{G}_0^e|$. Since $|\mathcal{G}_0^e|$ is a static value, minimizing $delay(\Gamma) = \frac{1}{|\mathcal{G}_0^e|} flowtime(\Gamma)$ is obvious equivalent to minimizing $flowtime(\Gamma)$.

Proposition 3. Minimizing average travel distance in online scenarios is equivalent to minimizing sum-of-costs in offline scenarios.

Proof. As defined in the main text,

$$dis(\Gamma) \doteq \frac{1}{|\mathcal{A}_0^e| + \sum_{t=1}^{T-1} |\mathcal{A}_t^{e+}|} \sum_{t=1}^{T} \sum_{a \in \mathcal{A}_{t-1}^e} \mathbf{1}(p_{t-1}^e(a) \neq p_t^b(a)) \tag{A5}$$

$$soc(\Gamma) \doteq \sum_{t=1}^{T} \sum_{a \in \mathcal{A}_{t-1}^e} \mathbf{1}(p_{t-1}^e(a) \neq p_t^b(a)) \tag{A6}$$

In offline scenarios, no agents are added after time step 0, i.e., $|\mathcal{A}_0^e| + \sum_{t=1}^{T-1} |\mathcal{A}_t^{e+}| = |\mathcal{A}_0^e|$. Since $|\mathcal{A}_0^e|$ is a static value, minimizing $dis(\Gamma) = \frac{1}{|\mathcal{A}_0^e|} soc(\Gamma)$ is obvious equivalent to minimizing $soc(\Gamma)$.

Appendix B Method Details

The training procedure of PROFILE is illustrated by Algorithm 1.

* Corresponding author (email: zhangw.sei@pku.edu.cn, zhhy.sei@pku.edu.cn, zhijin@pku.edu.cn)

Algorithm 1: The training procedure of PROFILE

Input: $DT = \{(\Omega_{e,t}, a_i, y^{Opt}_{e,t,i}) | \Omega_{e,t} \in \Omega, a_i \in \mathcal{A}, y^{Opt}_{e,t,i} \in \mathcal{Y} \wedge y^{Opt}_{e,t,i} = VOGS(\Omega_{e,0}, a_i, t)\}$: training data generated by VOGS algorithm, where $VOGS$ function return the action of agent a_i at time t along its path planned by VOGS algorithm for the e^{th} experiment;

While not done **do**
  Sample a batch of samples DT_i from DT
  With perception field learner **do**
    Compute interaction matrices IM with their definition
    Compute distance embeddings D^{Man}, D^{Hor} and D^{Ver} with Equation 9 and Equation 10
    Compute the perception field $\mathcal{F}^*$ with Equation 8 and Equation 11
  With integrating field classifier **do**
    Compute $query^{Hor}$, $query^{Ver}$, key^{Hor}, key^{Ver}, W^{Hor} and W^{Ver} with Equation 12
    Compute $value^{Hor}$, $value^{Ver}$, $Z^{Hor}_{cls'}$ and $Z^{Hor}_{cls'}$ with Equation 13
    Compute the final prediction $P_\Phi(Y|\Omega, a)$ with Equation 14 and Equation 15
  Compute the cross-entropy loss l with Equation 16
  Update all the learnable parameters $\theta \leftarrow \theta - \eta \bigtriangledown_\theta l(\theta)$
end while

Appendix C Evaluation Metrics

Our approach aims to learn an effective solution for both offline and online unlabeled MAPF tasks with a large collective. Hence, to research generalization ability, we evaluate the average completion rate, success rate, and average sum-of-costs in offline settings, and average throughput in online settings; to research scalability, we evaluate average running time in both offline and online settings. Thus we give the definitions of these metrics as follows.

Metrics for Generalization Ability.

Offline Metric. Given an offline test setting $E = (G_{[W,H]}, A, S, \mathcal{T}, T, N)$ where $G_{[W,H]}$ denotes a grid environment; A denotes a group of agents; S denotes a target shape; $\mathcal{T}$ denotes a running time threshold; T denotes a moving steps threshold; and N denotes the number of repeated experiments for A to form S within the threshold $\mathcal{T}/T$ in environment $G_{[W,H]}$, the effectiveness of a policy $\mathcal{P}$ under this setting E is evaluated by three indicators: average completion rate, success rate, and average sum-of-costs.

Definition 1 (Avg. Completion Rate)**.** The average completion rate, i.e., the average rate of target grids to be occupied at the end of each experiment, is denoted as ρ and defined as follows:

$$\rho(E, \mathcal{P}) \doteq \frac{1}{N} \sum_{e=1}^{N} \frac{|C_{E,\mathcal{P},e}|}{|S|}, \tag{C1}$$

where $C_{E,\mathcal{P},e}$ denotes the set of occupied target grids when the e'th experiment terminates.

Definition 2 (Success Rate)**.** The success rate, i.e., the rate of completely forming the shape in N experiments, is denoted as ζ and defined as follows:

$$\zeta(E, \mathcal{P}) \doteq \frac{1}{N} \sum_{e=1}^{N} \mathbf{1}(e), \tag{C2}$$

where $\mathbf{1}(e) = 1$ if S is formed when the e'th experiment terminates, and 0 otherwise (the experiment terminates when either S is formed, or the number of iterations exceeds a pre-defined threshold T, or the running time exceeds a pre-defined threshold $\mathcal{T}$).

Definition 3 (Avg. Sum-of-Costs)**.** A commonly used measure average sum-of-costs [4, 9], i.e., the average total travel distance of the group in each experiment, is denoted as l and defined as follows:

$$l(E, \mathcal{P}) \doteq \frac{1}{\sum_{e=1}^{N} \mathbf{1}(e)} \sum_{e=1}^{N} \mathbf{1}(e) \sum_{i=1}^{|A|} |p_{i,E,\mathcal{P},e}|, \tag{C3}$$

where $p_{i,E,\mathcal{P},e}$ denotes the path of agent a_i to form shape S under setting E by algorithm $\mathcal{P}$ when the e'th experiment terminates.

Online Metric. Referring to works in [7, 10], we research two online scenarios, called *pick-up* and *intersection*. In both online scenarios, N targets and N agents are randomly distributed in a given grid environment at the beginning. In pick-up(/intersection), when an agent arrives at a target grid, the current target(/agent) disappears, and a new target(/agent) will be randomly released at an empty grid, i.e., a grid without any agent and task. All agents need to reach and complete targets within the environment continuously. The objective is to maximize the throughput, i.e., the average number of targets reached per unit of time.

Accordingly, given a pick-up(/intersection) setting $E^d = (G_{[W,H]}, A_0, S_0, \mathcal{T}, T, N)$, where A_0 and S_0 are the initial agent and target locations in the scenario, the scenario generality of a policy $\mathcal{P}$ under this setting is evaluated by the average throughput.

Definition 4 (Avg. Throughput)**.** The average throughput is denoted as o and defined as follows:

$$o(E^d, \mathcal{P}) \doteq \frac{1}{N \times T} \sum_{e=1}^{N} \sum_{t=1}^{T} |I_t|, \tag{C4}$$

where $I_t = \{a_i | p_t(a_i) \in S_t\}$ denotes the set of agents who reaches a target grid at time t, $p_t(a_i)$ indicates the location of a_i at time t.

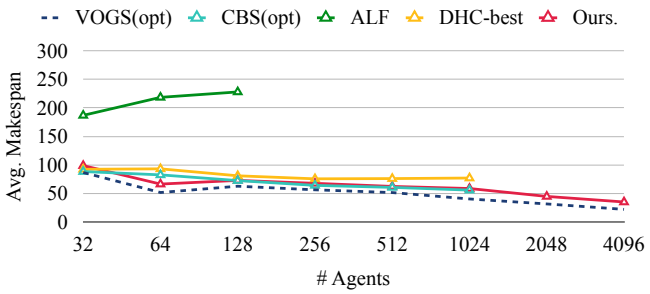

Figure E1 The avg. makespan under random target distribution in 160×160 world.

Metrics for Scalability.

Offline Metric. Given an offline test setting $E = (G_{[W,H]}, A, S, \mathcal{T}, T, N)$, the efficiency of a policy $\mathcal{P}$ under this setting is evaluated by the average running time (per case).

Definition 5 (Avg. Running Time (Offline))**.** The average running time is denoted as τ and defined as follows:

$$\tau(E, \mathcal{P}) \doteq \frac{1}{N} \sum_{e=1}^{N} \tau_{E,\mathcal{P},e}, \tag{C5}$$

where $\tau_{E,\mathcal{P},e}$ denotes running time of algorithm $\mathcal{P}$ to form shape S under setting E when the e'th experiment terminates.

Online Metric.

Given a pick-up(/intersection) setting $E^d = (G_{[W,H]}, A_0, S_0, \mathcal{T}, T, N)$, the efficiency of a policy $\mathcal{P}$ under this setting is evaluated by the average running time (per step).

Definition 6 (Avg. Running Time (Online))**.** The average running time in an online scenario is denoted as ϵ and defined as follows:

$$\epsilon(E^d, \mathcal{P}) \doteq \frac{1}{T} \tau(E^d, \mathcal{P}). \tag{C6}$$

Appendix D Baselines

MAPF baselines.

(1) CBS [1], a centralized distance-optimal method for multi-agent pathfinding (MAPF) with a given task assignment, expands the search tree by adding new constraint nodes when encountering conflicts. The code is available at https://github.com/Jiaoyang-Li/CBSH2-RTC.

(2) DHC [7], a reinforcement learning based algorithm for MAPF with a given task assignment, treats each agent independently and embeds the potential choices of shortest paths from a single source as heuristic guidance. The code is available at https://github.com/ZiyuanMa/DHC.

(3) PRIMAL$_2$ [3], a distributed learning based algorithm combining reinforcement learning and imitation learning for one-shot MAPF and life-long MAPF (LMAPF). ODrM* is used to generate expert demonstrations. Two models, PRI$_2$-M and PRI$_2$-LM, are trained in MAPF and LMAPF scenarios, respectively. The code is available at https://github.com/marmotlab/PRIMAL2.

(4) MAGAT [5], an imitation learning based approach that utilizes a message-aware graph attention network to learn from an expert for the MAPF problem. The code is available at https://github.com/proroklab/magat_pathplanning.

To the best of our knowledge, there is no learning-based approach for static/online unlabeled MAPF in the grid world, which is why we choose three learning-based methods (DHC, PRIMAL$_2$, and MAGAT) designed for MAPF as our baselines. Among them, MAGAT is an IL approach and has a very similar setting to PROFILE.

Unlabeled MAPF baselines.

(1) VOGS [11], a centralized distance-optimal method for shape formation, utilizes a vertex ordering and goal-swap policy in path replanning to resolve conflicts. We also use VOGS to generate train/test datasets for supervised learning. Code is implemented by ourselves, according to the paper.

(2) ALF [2], a self-organized method based on an artificial light field for shape formation, designs two different light signals to attract agents towards targets and expel agents away from others. Code is implemented by ourselves, according to the paper.

(3) TSWAP [8] is a complete centralized algorithm for unlabeled MAPF, consisting of target assignment with lazy evaluation and path planning with target swapping. The code is available at https://github.com/AlbaIntelligence/unlabeled-MAPF.

(4) Maxflow [6, 12] reduces unlabeled MAPF into a maxflow problem and solves it within polynomial time. The code is available at https://github.com/AlbaIntelligence/unlabeled-MAPF.

Appendix E Other Results

Appendix E.1 Results on Makespan

We also evaluate the average makespan, i.e., the average maximum travel distance of the group in each experiment, which is denoted as t and defined as follows:

Definition 7 (Avg. Makespan)**.**

$$t(E, \mathcal{P}) \doteq \frac{1}{\sum_{e=1}^{N} \mathbf{1}(e)} \sum_{e=1}^{N} \mathbf{1}(e) \max_{i \in [1,|A|]} |p_{i,E,\mathcal{P},e}|. \tag{E1}$$

Table E1 The changes of avg. makespan (*Mksp*) under random target distribution with 512 agents and increasing world sizes. (The best results among all planning-based/learning-based methods are bolded, and the best results among all methods are underlined.)

	World Size	Planning-Based Methods					Learning-Based Method			
		VOGS	CBS	TSWAP	Maxflow	ALF	DHC	PRI_2-M	MAGAT_b	Ours.
Mksp	40	10	-	5.7	**4.4**	-	-	-	-	**11.1**
	80	23.8	28.9	10.9	**10.0**	-	54.9	-	-	**29.8**
	160	40.5	61.0	21.9	**21.5**	-	77.2	-	-	**53.6**

Table E2 The changes of avg. makespan (*Mksp*) under 156 types of connected target distribution with increasing world sizes. (The best results among all planning-based/learning-based methods are bolded, and the best results among all methods are underlined.)

	World Size	Planning-Based Methods					Learning-Based Method			
		VOGS	CBS	TSWAP	Maxflow	ALF	DHC	PRI_2-M	MAGAT_b	Ours.
Mksp	10	6.5	7.8	6.0	**5.7**	15.6	14.1	-	-	**8.5**
	20	14.2	16.8	12.9	**11.8**	28.8	31.6	-	-	**22.6**
	40	31.5	36.5	30.6	**24.4**	49.2	-	-	-	**40.8**

The avg. makespan under random target distribution of different methods in 160×160 world is shown in Figure.E1. The PROFILE's performance on avg. makespan under random target distribution with 512 agents and increasing world sizes is shown in Table. E1. And the PROFILE's performance on avg. makespan under 156 types of connected target distribution with increasing world sizes is shown in Table. E2.

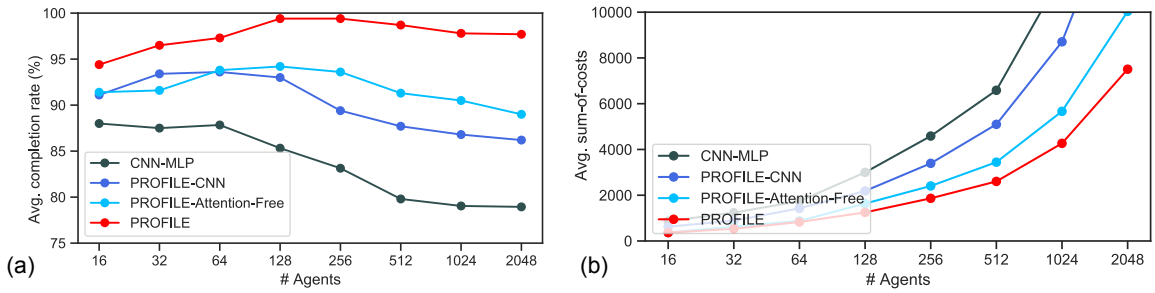

Figure E2 The ablation study results under random target distribution in offline 80*times*80 world as the team size increases. (a) the avg. completion rate of CNN-MLP, PROFILE-CNN, PROFILE-Attention-Free, and PROFILE; (b) the avg. sum-of-costs of CNN-MLP, PROFILE-CNN, PROFILE-Attention-Free, and PROFILE

Appendix E.2 Results for Ablation Study

This section reports the results of ablation experiments. Specifically, we test the three variants of PROFILE (PROFILE-CNN, PROFILE-Attention-Free, and CNN-MLP) in two offline settings and two online settings like experiments in the main text Section 5.3.1.

For offline experiments, the results under random target distribution are reported in Appendix Table E3 and Figure E2. Comparing results in Table E3 and Figure E2 with the results in Figure 4.(c), we can find that when the team density remains the same (e.g., 128 agents in a 40-sized world v.s. 512 agents in an 80-sized world), the performances of PROFILE and PROFILE-Free-Attention are relatively consistent, while the performances of PROFILE-CNN and CNN-MLP sharply drop. Such results reflect that the perception field has a greater effect and necessity in larger environments.

The results under connected target distribution are reported in Appendix Table E4. Without the perception field, the performances of two variants (PROFILE-CNN and CNN-MLP) sharply decrease in both the average completion rate and success rate. However, these two variants still show an advantage over learning-based baselines like DHC in the main text. Two possible reasons behind this are (1) the differences between task assignment policy in training and testing. Specifically, current learning-based methods are designed for labeled MAPF instead of unlabeled MAPF; thus, the trained models in these methods rely on the properties of the hidden task assigner behind its training data consisting of specific task assignment information. And when the task assignment policies are mismatched between training and testing, the method's performance will drop; (2) the suboptimality of decoupled task assignment and pathfinding in unlabeled MAPF. As mentioned in related works in the main text, simply decomposing unlabeled MAPF into two independent sub-problems of task assignment and pathfinding seriously affects the quality and efficiency of the final solution since these two sub-processes are not jointly optimized.

The results of two online settings are reported in Appendix Table E5, in which all variants achieve acceptable performance.

In summary, ablation results show the following three patterns:

Table E3 The avg. completion rate (CR) and avg. sum-of-costs (SoC) of PROFILE and its variants under random target distribution in 80-size world with increasing number of agents.

	#Agt.	PROFILE	CNN-MLP	PR.-CNN	PR.-Att.Free
	16	94.4	88.0	91.1	91.4
	32	96.5	87.5	93.4	91.6
	64	97.3	87.8	93.6	93.8
	128	99.4	85.3	93.0	94.2
CR	256	99.4	83.1	89.4	93.6
	512	98.7	79.8	87.7	91.3
	1024	97.8	79.0	86.8	90.5
	2048	97.7	78.9	86.2	89.0
	16	364.0	866.3	622.4	373.2
	32	533.3	1237.1	890.5	619.9
	64	828.5	1739.9	1425.0	877.3
	128	1248.7	2996.9	2185.2	1635.6
SoC	256	1864.2	4585.9	3392.8	2404.0
	512	2602.9	6585.2	5097.5	3444.0
	1024	4267.2	11649.5	8705.1	5666.8
	2048	7507.3	20494.9	15990.5	10035.4

Table E4 The avg. completion rate (CR), success rate (SR), and avg. sum-of-costs (SoC) of PROFILE and its variants under 156 types of connected target distribution with increasing world sizes.

	Size	PROFILE	CNN-MLP	PR.-CNN	PR.-Att.Free
	10	100.0	74.5	76.0	99.6
CR	20	99.9	66.9	70.7	98.9
	40	99.7	44.0	58.5	99.4
	10	100.0	70.3	72.3	94.1
SR	20	97.2	61.5	66.4	91.5
	40	92.6	28.0	31.7	90.7
	10	89.9	175.8	142.0	92.4
SoC	20	532.3	1065.5	776.1	606.1
	40	3732.3	6213.6	5455.6	3855.0

• in both static and dynamic settings, the perception contributes significantly to PROFILE's generalization ability, and the triplet cross-attention mechanism is also indispensable because it helps to further integrate and refine the perception fields and utilize them for high-quality action decision-making;

• comparing two static settings, the existence of perception field has a greater effect on the method's performance under connected than random target distributions scenarios, while the opposite is true for the triplet cross-attention mechanism;

• comparing two dynamic settings, there is no apparent difference between the effects of the perception field in pick-up and intersection scenarios, and the same is to the triplet cross-attention mechanism.

Table E5 The avg. throughput (TP) of PROFILE and its variants in 80-size world pick-up online scenarios and 160-size world intersection online scenarios with increasing number of agents.

	#Agt.	**PROFILE**	**CNN-MLP**	**PR.-CNN**	**PR.-Att.Free**
	16	0.7	0.5	0.5	0.6
	32	2.2	1.4	1.5	1.9
	64	6.7	3.8	4.1	6.1
	128	19.2	12.1	13.4	19.7
Pick.TP	256	55.3	38.6	40.4	44.1
	512	159.0	130.1	137.1	119.8
	1024	458.5	325.2	390.4	421.5
	2048	1310.9	862.3	904.6	1071.0
	64	3.6	1.5	1.9	3.5
	128	9.5	3.8	5.8	7.5
	256	25.9	12.1	15.8	25.1
	512	75.7	37.0	50.3	63.3
Int.TP	1024	217.6	108.8	145.2	232.4
	2048	622.5	356.7	421.3	525.3
	3072	1159.6	697.8	689.5	1132.5
	4096	1765.1	1085.7	1136.9	1289.6
	5120	2464.3	1436.7	1593.7	2445.7
	6144	3259.2	2315.2	2527.7	2997.9
	7168	4114.5	2482.7	2886.8	3861.3

References

1 Eli Boyarski, Ariel Felner, Roni Stern, Guni Sharon, David Tolpin, Oded Betzalel, and Eyal Shimony. Icbs: Improved conflict-based search algorithm for multi-agent pathfinding. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

2 Wenjie Chu, Wei Zhang, Haiyan Zhao, Zhi Jin, and Hong Mei. Massive shape formation in grid environments. *IEEE Transactions on Automation Science and Engineering*, pages 1–15, 2022.

3 Mehul Damani, Zhiyao Luo, Emerson Wenzel, and Guillaume Sartoretti. Primal_2: Pathfinding via reinforcement and imitation multi-agent learning-lifelong. *IEEE Robotics and Automation Letters*, 6(2):2666–2673, 2021.

4 Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 31:591–656, 2008.

5 Qingbiao Li, Weizhe Lin, Zhe Liu, and Amanda Prorok. Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robotics and Automation Letters*, 6(3):5533–5540, 2021.

6 Hang Ma and Sven Koenig. Optimal target assignment and path finding for teams of agents. In Catholijn M. Jonker, Stacy Marsella, John Thangarajah, and Karl Tuyls, editors, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 1144–1152. ACM, 2016.

7 Ziyuan Ma, Yudong Luo, and Hang Ma. Distributed heuristic multi-agent path finding with communication. *arXiv preprint arXiv:2106.11365*, 2021.

8 Keisuke Okumura and Xavier Défago. Solving simultaneous target assignment and path planning efficiently with time-independent execution. *arXiv preprint arXiv:2109.04264*, 2021.

9 Trevor Standley. Finding optimal solutions to cooperative pathfinding problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

10 Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Barták, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In Pavel Surynek and William Yeoh, editors, *Proceedings of the Twelfth International Symposium on Combinatorial Search, SOCS 2019, Napa, California, 16-17 July 2019*, pages 151–159. AAAI Press, 2019.

11 J. Yu and S. M. LaValle. Shortest path set induced vertex ordering and its application to distributed distance optimal formation path planning and control on graphs. In *52nd IEEE Conference on Decision and Control*, pages 2775–2780, 2013.

12 Jingjin Yu and Steven M LaValle. Multi-agent path planning and network flow. In *Algorithmic foundations of robotics X*, pages 157–173. Springer, 2013.