Plan merging

Aurélien SIMON

June 21, 2022

Abstract

We are talking about plan merging here

1 Definition

Plan merging aim to solve MAPF [1, 2] problems by computing individual plan for each agent to their goal regardless of conflict, and then, by using these previous plan, merge them into conflict-free plans. The task can be devided in three; target assigment, individual pathfinding and plan-merging.

1.1 Classic MAPF

We can base plan merging definition on classic MAPF's one; the paper "Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks" [2] provides a standart defintion for non-anonymous MAPF with k agents. And it is defined as such; it takes as input a tuple $\langle G, s, f\rangle$ where $G = (V, E)$ represent a graph. $s : [1, \dots, k] \to V$ maps an an agent to a "start" vertex. Finally, $f : [1, \dots, k] \to V$ maps an agent to a "final" vertex. The output is a set of k single-agent plan, where a plan π is denoted as a sequence of n action $a_n, \dots, a_1$, each of them being a function defined as $a : V \to V$, they denote a movement from a vertex to another iff an edge exist between these two or a wating if the two vertices are the same. Formally, $v \in V, v' \in V, \begin{cases} \exists e(v, v') \in E \\ v = v' \end{cases} \Rightarrow a(v) = v'$. Considering a plan π for an agent r and a timestep t, $\pi_r[t] = a_t(a_{t-1}(\dots a_1(s(r))))$. A plan is considered as valid iff $\pi_r[|\pi_r - 1|] = a_{|\pi_r|-1}(\dots(a_1(s(r)))) = f(r)$ where $|\pi_r|$ gives the length of the plan π_r. In order to have a valid solution, taken pairly, plans must be conflict-free;

1. a vertex conflict between two agents r and r' occurs if, at timestep t, $\pi_r[t] = \pi_{r'}[t]$.

2. an edge conflict (also called swapping conflict) between two agents r and r' occurs if, at timestep t, $\pi_r[t] = \pi_{r'}[t-1]$ and $\pi_r[t-1] = \pi_{r'}[t]$.

1.2 Target assignment

Basing ourselves on MAPF definition, we can then define target assignment (TA); it takes as input a tuple $\langle G, s, F\rangle$, $G = (V, E)$ representing a graph, $s : [1, \dots, k] \to V$ maps an agent to a "start" and F a set of k vertices denoting "final" vertices. The output would be a tuple $\langle G, s, f\rangle$ where $G = (V, E)$ representing a graph, $s : [1, \dots, k] \to V$ maps an agent to a "start" vertex. Finally, $f : [1, \dots, k] \to \{V\}$ maps an agent to a set of "final" vertices. Having a set of vertices as part of the output allows to describe both anonymous and non-anonymous MAPF problems, TA's output for anonymous variant would be the set of goal vertices F itself and a singleton for non-anonymous variant.

1.3 Individual Pathfinding

We would then define individual pathfinding (IP); the input of would be TA's output $\langle G, s, f\rangle$. IP would then give as output, $\langle G, \theta\rangle$ where each agent provides at least one solution. Formally, for each agent r, we have $\theta[r] = \{\pi_1, \ldots, \pi_n\}$. As a consequence of the TA's definition, let an agent r, $\forall \pi \in \theta[r], \pi[|\pi| - 1] \in f(r)$.

1.4 Plan merging

Finally, we can define plan merging (PM). It takes as input $\langle G, \theta \rangle$ since we can deduce start and final position from plans. And gives as output MAPF's one: a solution being a set of k conflict-free plans.

References

[1] E. Erdem et al. "A General Formal Framework for Pathfinding Problems with Multiple Agents". In: *Proceedings of the Twenty-Seventh National Conference on Artificial Intelligence (AAAI'13)*. Ed. by M. desJardins and M. Littman. AAAI Press, 2013, pp. 290–296.

[2] R. Stern et al. "Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks". In: *CoRR* abs/1906.08291 (2019). URL: `http://arxiv.org/abs/1906.08291`.