Plan merging

Aurélien SIMON

Potsdam University

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 a conflict-free plan. 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 "A General Formal Framework for Pathfinding Problems with Multiple Agents"[1] provides a standart defintion for non-anonymous MAPF. 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 action $a_1, \dots, a_n$, each of them being a function defined as $a : V \to V$. Plans must satisfy these properties:

1. considering an agent r and its plan π_r, $\pi_r[|\pi_r|] = f(a_n(\dots(a_1(s(r)))))$
2. do not contain any vertex conflict. Vertex conflict exist between π_i and π_j iff for any time step t, $\pi_i[t] = \pi_j[t]$.
3. do not contain any edge conflict. Edge conflict exist between π_i and π_j iff for any time step t, $\pi_i[t+1] = \pi_j[t]$ and $\pi_i[t] = \pi_j[t+1]$.

1.2 Target assignment

Basing ourselves on MAPF definition, we can then define target assignment (TA)[1][3]; 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 "final" vertex.

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 $\theta[r] = \{\pi_1, \dots, \pi_n\}$. Plans contained in θ must only satisfy property 1.

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 plan π.

References

1. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. Written by Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Eli Boyarski, Roman Barták
2. A General Formal Framework for Pathfinding Problems with Multiple Agents. Written by Esra Erdem, Doga G. Kisa, Umut Oztok and Peter Schüller
3. Structure and Intractability of Optimal Multi-Robot Path Planning on Graphs. Written by Jingjin Yu and Steven M. LaValle