Scalable Iterative Multi-Agent Path Finding under Turn Constraints (Supplementary File)

Appendix A
Detailed complexity analysis

This section provides detailed complexity analysis of proposed algorithm.

As shown in Algorithm 1, within a single timestep of planning, sorting the agents based on updated priorities (line 2) requires $O(|A|lg|A|)$. During the planning process for each agent, iterating over the candidate nodes requires $O(\Delta(G)+1)$, where $O(\Delta(G))$ is the maximum degree of the graph. Both detecting cycles and handling cycles requires $O(|C|)$, where $|C|$ is the number of agents in the cycle with $|C| \leq |A|$, resulting in an overall complexity of $O(|A|)$. Assuming other operations are completed in constant time $O(K)$, the proposed algorithm achieves a time complexity of $O(|A|\cdot(lg|A|+|A|\Delta(G)+K))$. It is evident that the proposed method requires only polynomial time complexity for planning within a single timestep, offering a significant computational advantage over algorithm with exponential time complexity.

The time complexity for creating the cost table during initialization is $O(4|A|\cdot(4|V|\cdot 3)) = O(|A|\cdot|V|)$. In large-scale instances involving extensive maps or a high number of agents, the preprocessing time will increase accordingly. However, experiments on benchmark maps have demonstrated acceptable preprocessing speeds, which are further explained in the numerical experiments section.

Appendix B
Proof of lemma in Theorem 1

This section provides the formal proof for the distance property $d(s,g) = \min_{a\in A}\{d(s,a)+d(a,g)\}$ utilized in Theorem 1 to establish the optimality of the Local Greedy Strategy.

Lemma 1: If every shortest path from node s to g passes through some node $a \in A$, then: $d(s,g) = \min_{a\in A}\{d(s,a) + d(a,g)\}$.

Proof: Let P be a shortest path from s to g with cost $d(s,g)$. By assumption, P must pass through some node $a^* \in A$. By the optimal substructure property of shortest paths, any subpath of a shortest path is also a shortest path. Thus:

$$d(s,g) = d(s,a^*) + d(a^*,g) \geq \min_{a\in A}\{d(s,a)+d(a,g)\}.$$

Let $a' = \arg\min_{a\in A}\{d(s,a)+d(a,g)\}$. The path $s \to a' \to g$ has length $d(s,a')+d(a',g)$. Since $d(s,g)$ is the shortest path length:

$$d(s,g) \leq d(s,a') + d(a',g) = \min_{a\in A}\{d(s,a)+d(a,g)\}.$$

Combining both inequalities gives

$$d(s,g) = \min_{a\in A}\{d(s,a)+d(a,g)\}.$$

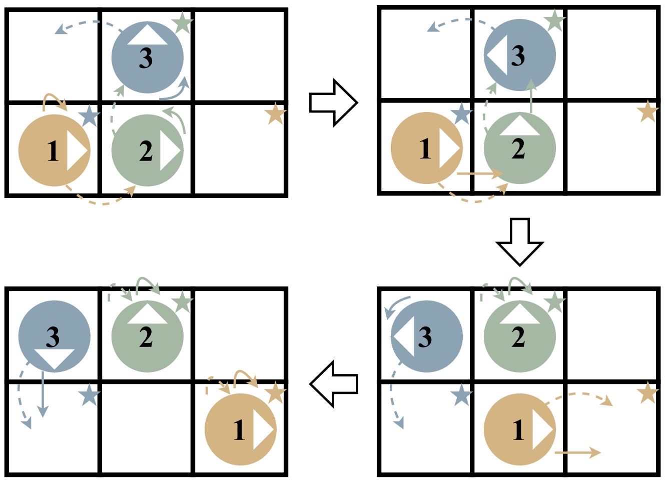

Fig. A1: An instance solved by PIBT_T^*. The final step is omitted.

Appendix C
An illustrative example of PIBT_T^*

This section illustrates a case study using the proposed method as shown in Fig. A1. The example involves three agents, a_1, a_2, and a_3, on a 2×3 grid with priorities assigned as $a_1 > a_2 > a_3$. In the figure, goal nodes for each agent are represented by stars of the corresponding colors. Dashed arrows indicate successful node requests for the current timestep, while solid arrows represent the actual actions executed by the agents.

At the initial timestep, the algorithm first plans for a_1 as the highest-priority agent. Agent a_1 requests the node currently occupied by a_2. Because the degree of current node of a_2 is greater than 2, the swap procedure for a_1 is not triggered, and the recursive $\text{PIBT}_\text{T}^*(a_2, a_1)$ procedure is invoked. Subsequently, a_2 requests the node occupied by a_3. Similarly, the swap procedure for a_2 is not initiated because the degree of current node of a_3 exceeds 2, leading to the recursive call of $\text{PIBT}(a_3, a_2)$. Agent a_3 then attempts to request the node to its left, which is successful since the node is currently available.The action update function then determines the first action for a_3 to reach the requested node, which consists of a 90° counterclockwise rotation. This request and action information are then returned to a_2. Following the same logic, a_2 also executes a 90° counterclockwise rotation to move toward its requested node and returns its information to a_1. Although a_1 successfully requests the node to its right and is already oriented toward that direction, it is blocked by a_2 who is performing an in-place rotation. Consequently, the action update function modifies the planned forward move of a_1 to a wait action for the current timestep.

In the next timestep, the node request process remains the same to the previous step because the positions of all agents remain unchanged. During this timestep, a_3 is capable of executing a forward move to the left. Once a_3 returns its action information to a_2, a_2 is also able to move upward, and a_1 proceeds to the right. As a result, the three agents sharing the same temporary priority collectively complete their movements.

During the third timestep, the algorithm continues to plan actions for the agents according to the priority order. Agent a_1 requests the goal node located on the right. This node is unoccupied and a_1 is already oriented toward the goal node. Therefore, a_1 can move to the destination in the next timestep. Since a_2 is already at the goal node, it requests the current node and plans a wait action. Meanwhile, a_3 requests the node below and executes a 90° counterclockwise rotation.

From the fourth timestep onward, both a_1 and a_2 have reached the respective goal nodes, while a_3 requires only one more timestep to arrive at the target. The instance is solved at the fifth timestep.

Appendix D
Selection of k value

This section presents the complete experimental data plots regarding the success rate, sum of costs, and makespan of PIBT$^*_\mathrm{T}$ across various values of k, as Fig. A2 shows.

The configuration $k = \infty$ is equivalent to the omission of the PET module during the node selection process.

Appendix E
Ablation experiment of SPD and PET

In the main text, Section IV(C) presents the success rate and makespan data for the *random* and *warehouse* maps as the number of agents increases.

The complete data plots for the ablation experiment of the SPD and the PET across all five maps are illustrated in Fig. A3.

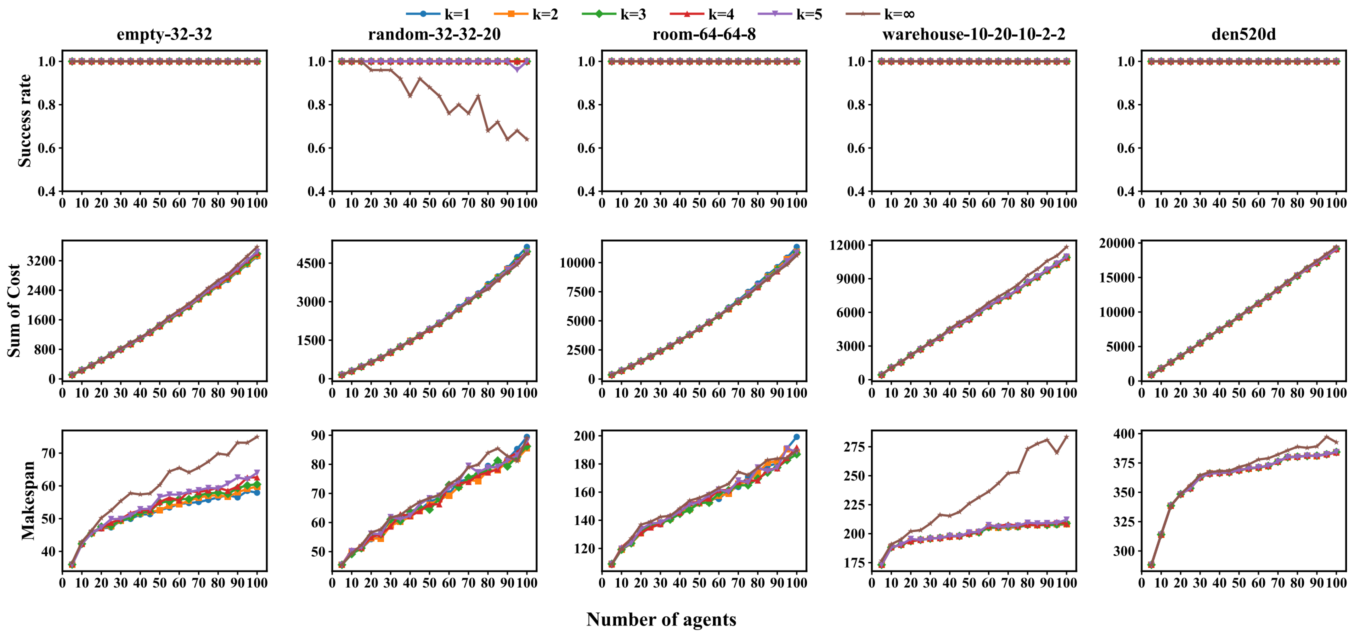

Fig. A2: Success rate, sum of costs and makespan of PIBT$_T^*$ with different k values.

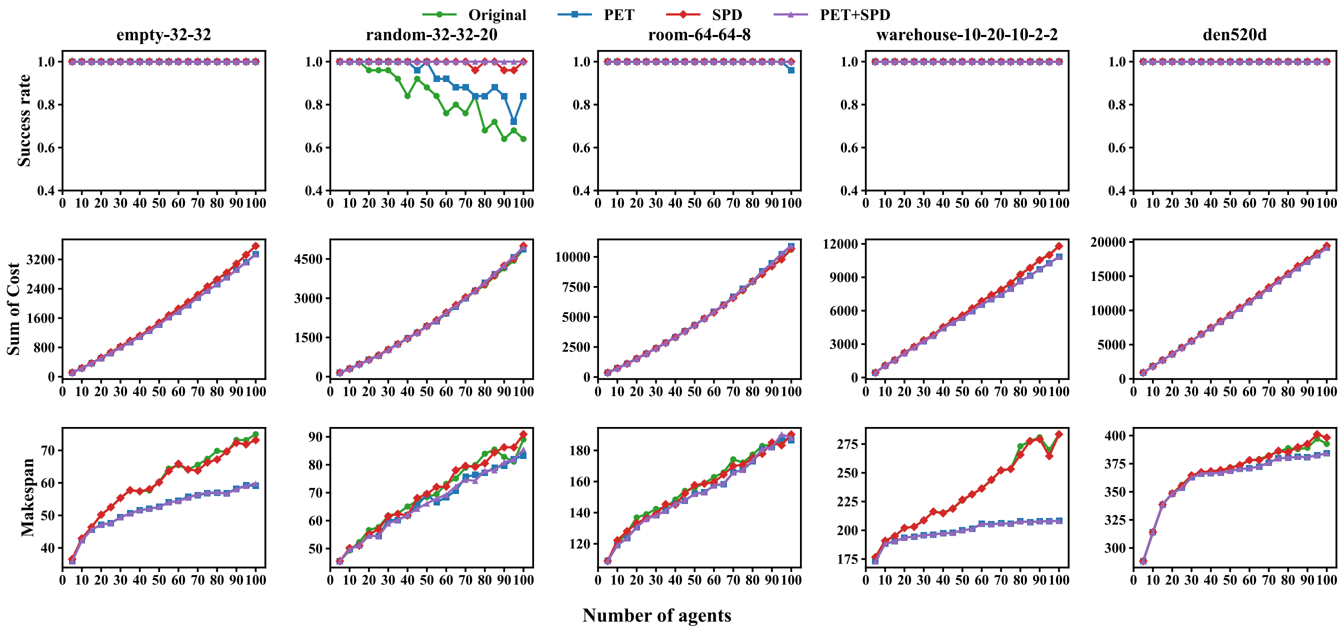

Fig. A3: Ablation experiment of SPD and PET in 5 different maps.