7 March 2025 ·

Qeios

Peer Review

Review of: "Enhancing Lifelong Multi-Agent Path Finding with Cache Mechanism"

Suhaib Alansary[1]

1. Computer Science, University of Basrah, Iraq

(Review of "Enhancing Lifelong Multi-Agent Path Finding with Cache Mechanism)

This paper introduces L-MAPF-CM, a novel framework integrating cache grids and a task assigner (TA) with a lock mechanism into Lifelong MAPF (L-MAPF) to improve warehouse automation efficiency. The cache mechanism reduces travel time for frequently accessed items, while the TA manages agent workflows and cache interactions. Experiments demonstrate performance improvements in throughput, particularly under high cache hit rates, though traffic congestion at scale remains a challenge.

1. The task assigner is simple; suggesting a more sophisticated one could improve results.

2. LaCAM's congestion issues under high agent density might affect the validity; maybe test with other solvers.

3. The lock mechanism's overhead isn't discussed. How does it impact performance?

4. The paper claims to be the first to combine cache with MAPF, but is there any related work in robotics using similar concepts?

5. The experiments use synthetic and real data, which is good, but more details on real data setup would help reproducibility.

6. The impact of cache replacement policies is minimal in results; perhaps explore why or suggest better policies.

7. Traffic congestion is a major factor; discuss potential solutions or how cache could be positioned to alleviate this.

 doi.org/10.32388/HF77DS

8. The figures are referenced but not included, making it hard to assess. Ensure they are clear in the final version.

9. The deadlock-free proof is in the appendix? It's mentioned but not detailed in the method section.

10. The supplementary materials (code, video) are crucial; ensure they are accessible.

Weaknesses and Suggestions for Improvement

1. **Task Assigner Simplicity**:

- The TA's "greedy" assignment policy (allocating tasks one-by-one) may limit performance. Consider integrating predictive or learning-based strategies to optimize task ordering and cache utilization.

- **Suggestion**: Explore task batching or priority-based assignment to reduce agent idle time and improve cache hit rates.

2. **Dependency on LaCAM**:

- Traffic congestion under high agent density (e.g., 256 agents) highlights LaCAM's limitations in dense scenarios. This risks conflating solver-specific issues with the cache mechanism's efficacy.

- **Suggestion**: Benchmark with other L-MAPF algorithms (e.g., RHCR, PIBT) to isolate the cache mechanism's impact.

3. **Lock Mechanism Overhead**:

- The paper does not quantify the computational cost of lock acquisition/release. This could affect real-time performance in large-scale systems.

- **Suggestion**: Include metrics on lock contention and latency, especially in high-agent scenarios.

4. **Cache Replacement Policies**:

- Results show minimal differences between LRU, FIFO, and RANDOM. Clarify whether this is due to task distribution properties or inherent policy limitations.

- **Suggestion**: Test adaptive policies (e.g., learning-based replacements) or hybrid strategies tailored to warehouse task patterns.

5. **Traffic Congestion**:

- Congestion near caches/unloading ports (Figure 5) diminishes gains at scale. While mentioned, solutions are deferred to future work.

doi.org/10.32388/HF77DS

- **Suggestion**: Discuss immediate mitigations (e.g., one-way lanes, dynamic cache allocation) in the context of the current framework.

6. **Real-World Data Details**:

- The "Real Data Distribution" is under described. Clarify how the Kaggle dataset was processed and whether temporal/spatial task correlations were preserved.

- **Suggestion**: Provide dataset preprocessing steps and task generation methodology for reproducibility.

7. **Comparison with Competing Approaches**:

- The baseline is vanilla L-MAPF. Compare with other cache-aware warehouse strategies (e.g., turnover-based storage) to contextualize improvements.

8. **Deadlock Analysis**:

- The deadlock-free proof is briefly mentioned but not formalized. Include a formal proof or reference an appendix for rigor.

9. **Figures and Reproducibility**:

- Figures 4–7 are critical but absent in the review copy. Ensure visual clarity (e.g., color choices, axis labels) in the final version.

- **Suggestion**: Release code and datasets publicly to facilitate replication.

Additional Comments

- The paper's motivation (bridging high-level cache design and low-level planning) is compelling and addresses a real-world need.

- The lock mechanism's analogy to computer architecture is apt but could be expanded (e.g., how cache coherence protocols inspired the design).

- The distinction between single-port and multi-port scenarios is insightful but warrants deeper analysis (e.g., cross-port cache sharing).

Conclusion

The paper is solid but could benefit from addressing these points to strengthen the contribution and practical applicability.

I recommend this work to be accepted with minor corrections.

Declarations

Potential competing interests: No potential competing interests to declare.

doi.org/10.32388/HF77DS