Ghost Agents in SAT-based Models for Multi-Agent Pathfinding (Extended Abstract)

Jiří Švancara, Roman Barták
Charles University, Czech Republic
{svancara,bartak}@ktiml.mff.cuni.cz

Extended version — https://journals.flvc.org/FLAIRS/article/view/141799 (Švancara and Barták 2026)

Introduction

Multi-agent pathfinding (MAPF) is the task of navigating a set of agents from starting to goal locations while avoiding collisions (Stern et al. 2019). It has many practical applications, most notably in warehousing. Since finding an optimal solution is computationally hard, a popular approach is to reduce the problem to a well-established formalism such as SAT and invoke a high-performance solver.

Two competing SAT encodings exist for representing each agent's location. In the single-location encoding (Barták and Švancara 2019), an explicit constraint forces each agent to occupy exactly one vertex per time-step, yielding a unique path per agent. In the ghost agent encoding (Surynek et al. 2016), this constraint is omitted, allowing an agent to appear at multiple vertices simultaneously — so-called ghost agents. Removing this constraint reduces the formula size, potentially improving solving time. Notably, the original work does not explicitly discuss the presence of ghost agents or demonstrate that solutions remain correct. This paper addresses both gaps: we prove that ghost agents do not invalidate solutions and show how valid paths can be recovered via simple post-processing. We then empirically compare all combinations of (i) ghost vs. single encodings, (ii) eager vs. lazy conflict clause generation, and (iii) makespan vs. sum-of-costs objectives.

Problem Formulation and SAT Encoding

A MAPF instance is a pair (G, A) where $G = (V, E)$ is a graph and A is a set of agents, each with a start and a goal location. At each discrete time-step, agents may wait or move to an adjacent vertex. A valid joint plan must be collision-free: no two agents may share a vertex (vertex conflict) or traverse the same edge in opposite directions simultaneously (swapping conflict). Two cost objectives are commonly optimized: makespan (time until the last agent reaches its goal) and sum-of-costs (sum of individual arrival times). The SAT encoding introduces Boolean variables $At(a_i, v, t)$ and $Pass(a_i, u, v, t)$ to represent an agent's a_i presence at vertex v and traversal of edge (u, v) at time-step t, respectively. Unit clauses enforce start and goal locations. Movement constraints ensure agents depart through at least one edge and arrive at the correct vertex. Conflict constraints forbid vertex and swapping collisions between agent pairs. The single encoding additionally enforces *at-most-one* location per agent per time-step via an explicit constraint, whereas the ghost encoding omits this clause.

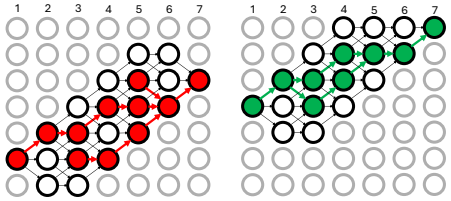

Figure 1: Solution example on a map of 7 vertices (vertical) in 7 time-steps (horizontal). Greyed vertices are unreachable. Red and green correspond to agents and phantom agents (i.e. At and $Pass$ set to $True$).

A limit on the number of time-steps is determined. In the case of a successful call to the solver, the found solution is optimal; otherwise, the limit is increased. This method guarantees makespan optimal solution. For sum-of-costs optimization, additional $Late(a_i, t)$ variables are introduced to count the number of time-steps an agent has not yet permanently settled at its goal. An *at-most-k* cardinality constraint bounds the total extra cost. All encodings exploit reachability preprocessing – variables and clauses are only created for positions that the agent can reach within the cost bound.

Ghost Agents

Without the at-most-one constraint, a satisfying assignment may place an agent at multiple vertices simultaneously (see Figure 1). We identify three ghost phenomena: (1) an agent may split when leaving a vertex by traversing more than one outgoing edge; (2) a ghost may appear out of nowhere since the movement implication constraints do not prevent unconstrained At variables from becoming true; and (3) ghost agents are still subject to all conflict constraints, so every true At value — real or ghost — must be collision-

free. We show that solution validity is preserved and that a genuine collision-free path can always be extracted via a backtrack-free depth-first search (DFS) starting from each agent's start location. Because any true At node in the time-expanded graph must eventually connect to the goal (enforced by the goal constraint and movement implications), the DFS is guaranteed to reach the goal regardless of which branch it follows when a ghost split occurs. Ghosts appearing out of nowhere are simply unreachable from the start and are ignored. The extracted paths are therefore valid MAPF solutions. For sum-of-costs optimization, a modified lateness constraint is required: instead of flagging an agent as late only when absent from its goal, it must be flagged whenever it appears anywhere outside its goal, preventing ghost agents at the goal from artificially zeroing the cost.

Conflict Encoding Strategies

Two strategies exist for encoding inter-agent conflict constraints. The eager approach adds all possible conflict clauses upfront (Barták and Švancara 2019), producing a larger but complete formula. The lazy approach (Surynek 2019) begins with no conflict clauses; once the SAT solver returns a (potentially conflicting) solution, the violated conflict constraints are added, and solving resumes incrementally. This exploits the incremental SAT solving paradigm while potentially avoiding generating many clauses.

Empirical Evaluation

We empirically compare eight encodings – the cross-product of {eager, lazy}, {single, ghost}, and {makespan, sum-of-costs} encodings. Experiments used grid maps of sizes 10×10 through 200×200 with three obstacle layouts: empty, random, and warehouse. For each map, 10 scenario files with randomly placed agents were generated. Instances contain 5 agents, incrementing by 5 until the time limit of 5 minutes is exceeded. Experiments were run on AMD EPYC 7543 core with 64 GB of memory. The encoding was implemented in C++ using the CaDiCaL SAT solver (Biere et al. 2024) and PBlib (Philipp and Steinke 2015) for cardinality constraints.

For *makespan* optimization, lazy encodings substantially outperform eager ones on larger maps due to smaller formula sizes. On small maps (20×20), however, eager encodings are superior: high agent density creates many conflicts that the lazy approach must discover iteratively, while the eager formula is not yet prohibitively large. A noteworthy interaction emerges between ghost and single variants: for eager encoding, the single variant performs better, while for lazy encoding, the ghost variant is stronger. We conjecture that the single-location constraint aids unit propagation within the eager SAT solver, whereas ghost agents generate additional inter-ghost conflicts in the lazy framework, causing earlier convergence to the necessary conflict clauses.

Under *sum-of-costs* optimization, performance differences among the four models are small. Eager variants solved marginally more instances than lazy ones, and ghost variants slightly outperformed their single counterparts. The sum-of-costs objective is inherently more restrictive about reachable agent positions at each time-step, leaving less room for ghost agents to arise. On larger maps, the lower agent density means paths rarely intersect, so few conflict clauses are needed regardless of the encoding strategy.

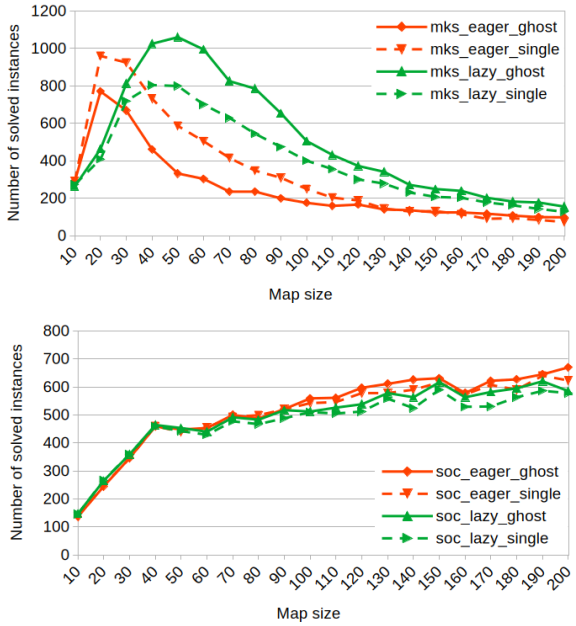

Figure 2: Number of solved instances under makespan (top) and sum-of-costs (bottom) objective.

Acknowledgments

Research supported by the CZ-IL Cooperative Scientific Research LUAIZ24104, and by Charles University UNCE 24/SCI/008. Presentation supported by the Horizon Europe Research and Innovation programme grant No. 101120406.

References

Barták, R.; and Švancara, J. 2019. On SAT-Based Approaches for Multi-Agent Path Finding with the Sum-of-Costs Objective. In *SoCS'19*, 10–17.

Biere, A.; Faller, T.; Fazekas, K.; Fleury, M.; Froleyks, N.; and Pollitt, F. 2024. CaDiCaL 2.0. In *CAV'24*, 133–152.

Philipp, T.; and Steinke, P. 2015. PBLib – A Library for Encoding Pseudo-Boolean Constraints into CNF. In *SAT'15*, 9–16.

Stern, R.; Sturtevant, N.; Felner, A.; Koenig, S.; Ma, H.; Walker, T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T.; Barták, R.; and Boyarski, E. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. In *SoCS'19*, 151–159.

Surynek, P. 2019. Unifying Search-based and Compilation-based Approaches to Multi-agent Path Finding through Satisfiability Modulo Theories. In *IJCAI'19*, 1177–1183.

Surynek, P.; Felner, A.; Stern, R.; and Boyarski, E. 2016. Efficient SAT Approach to Multi-Agent Path Finding Under the Sum of Costs Objective. In *ECAI'16*, 810–818.

Švancara, J.; and Barták, R. 2026. Ghost Agents in SAT-based Models for Multi-Agent Pathfinding. In *FLAIRS'26*.