Independence Detection in SAT-based Multi-Agent Path Finding

Pavel Surynek [*1]

Jiří Švancara [*2]

Ariel Felner [*3]

Eli Boyarski [*4]

[*1] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

[*2] Faculty of Mathematics and Physics Charles University, Prague, Czechia

[*3] Ben Gurion University Beer-Sheva, Israel

[*4] Bar-Ilan University Ramat-Gan, Israel

The problem of optimal multi-agent path finding (MAPF) is addressed in this paper. The task is to find optimal paths for mobile agents where each of them need to reach a unique goal position from the given start with respect to the given objective function. Agents must not collide with each other which is a source of combinatorial difficulty of the problem. An abstraction of the problem where discrete agents move in an undirected graph is usually adopted in the literature. Specifically, it is shown in this paper how to integrate independence detection (ID) technique developed for search based MAPF solving into a compilation-based technique that translates the instance of the MAPF problem into propositional satisfiability formalism (SAT). The independence detection technique allows decomposition of the instance consisting of a given number of agents into instances consisting of small groups of agents with no interaction across groups. These small instances can be solved independently and the solution of the original instance is combined from small solutions eventually. The reduction of the size of instances translated to the target SAT formalism has a significant impact on performance as shown in the presented experimental evaluation.

1. INTRODUCTION

Multi-agent path finding (MAPF) is the task is of finding collision free paths for a set of mobile agents so that each agent can reach its goal position from given start by following its path (Kornhauser et al., 1984, Silver, 2005, Surynek, 2009, Sharon et al. 2013). The MAPF problem recently attracted considerable attention from the research community and many concepts and techniques have been devised to address this problem.

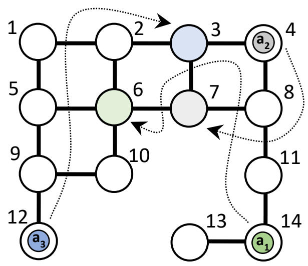

Figure 1: An example of a MAPF instance with three agents: a_1, a_2, and a_3.

An abstraction in which an environment with agents is represented by undirected graph is used in the literature (Wilson, 1974, Ryan, 2008). Agents in this abstraction are items placed in vertices of the graph. Edges represent passable regions. Physical space occupancy of agents is represented by the restriction that at most one agent can be placed in each vertex. The time is discrete which means that agents can do a single move in a time step. An example of the MAPF problem is shown Figure 1.

Typically, we are interested in two objectives: *makespan* and *sum-of-costs*. The makespan objective corresponds to time necessary until all agents reach their goals. While the sum-of-costs objective corresponds to the total number of actions needed to fulfill the goal or to intuitive consumption of energy. The presented techniques are generic across searching for optimal plans for both objectives.

The MAPF problem and its variants are strongly practically motivated. Applications range from navigation of multiple mobile robots (Berg et al., 2010), through traffic optimization (Kim et al., 2014), to movement planning in computer games (Wang, Botea, 2008). We refer the reader to various studies such as (Sharon et al. 2013, 2015) for the detailed survey of applications.

2. CONTRIBUTIONS

One of successful approaches for solving MAPF optimally is to translate the decision version into propositional formula (Surynek, 2014, Surynek et al., 2016) which is a technique inspired from the similar use of propositional satisfiability in the *classical planning* (Kautz et al., 1999, Huang et al., 2010).

The formula is satisfiable if and only if the instance of MAPF is solvable for a given value of the objective function. Assuming that satisfiability of such formula is a non-decreasing function of the value of objective function, it is easy to obtain the optimum by querying the satisfiability multiple times. A trivial strategy of increasing the value of objective function by one turned out to be the most efficient so far.

Corresponding author: Pavel Surynek, National Institute of Advanced Industrial Science and Technology (AIST), 2-3-26, Aomi, Koto-ku, Tokyo 135-0064, Japan.
pavel.surynek@aist.go.jp, +81 080 3495 1866

Satisfiability of the formula can be decided by an efficient off-the-shelf SAT solver (Biere et al., 2009, Audemard, Simon, 2009) which is one of the advantages of the SAT-based approach.

However, the most significant bottleneck of all the existing SAT-based algorithms for MAPF is the large size and combinatorial difficulty of the target propositional formula that grow significantly with the increasing number of agents as well as with growing size of the underlying graph.

This kind of growth of combinatorial difficulty has already been addressed by Standley (2010) in his search-based optimal MAPF solving algorithm. Standley described a method called *independence detection* (ID) that tries to determine the smallest possible groups of agents for which paths can be found independently of other groups. The ID technique turned out to be extremely beneficial when integrated with an algorithm for finding paths that has exponential time complexity in the number of agents. This is also the case of SAT-based MAPF solving.

Our contribution is integrating ID with MDD-SAT the most recent SAT-based MAPF solver (Surynek et al., 2016). As there are differences in how the original Standley's search-based algorithm and SAT-based approach work we suggested modifications to ID to be compatible with the SAT-based approach. Our new solver is called MDD-SAT+ID following the notation of Standley (2010). Conducted experiments demonstrate similar performance benefit as in the case of original application of ID. Considering that MDD-SAT has been state-of-the-art for a certain class of MAPF instances, the new MDD-SAT+ID represents new progress.

3. INDEPENDENCE DETECTION

We will first describe the original method of independence detection proposed by Standley (2010). The main idea behind this technique is that difficulty of MAPF solving optimally grows exponentially with the number of agents. It would be ideal, if we could divide the problem into a series of smaller sub problems, solve them independently, and then combine them.

The simple approach, called *simple independence detection* (SID), assigns each agent to a group so that every group consists of exactly one agent. Then, for each of these groups, an optimal solution is found independently. Every pair of these solutions is evaluated and if the two groups' solutions are in conflict (that is, when collision of agents belonging to different group occurs), the groups are merged and replanned together with the same value of the objective (this is a crucial step to ensure the optimality). If there are no conflicting solutions, the solutions can be merged to a single solution of the original problem. This approach can be further improved by avoiding merging of groups.

Generally, each agent has more than one possible optimal path. However, SID considers only one of these paths. The improvement of SID known as *independence detection* (ID) is as follows. Let's have two conflicting groups G_1 and G_2. First, try to replan G_1 so that the new solution has the same cost and the steps that are in conflict with G_2 are forbidden. If no such solution is possible, try to similarly replan G_2. If this is not possible, merge G_1 and G_2 into a new group. In case either of the replanning was successful, that group needs to be evaluated with every other group again. This can lead to infinite cycle. Therefore, if two groups were already in conflict before, we merge them without trying to replan.

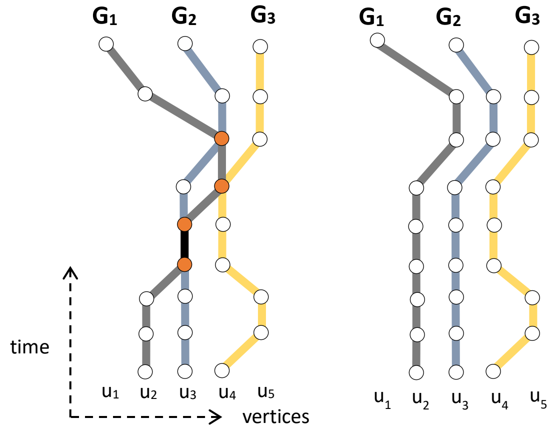

Figure 2: A schematic illustration of path replanning within the independence detection technique. A path for the group G_1 conflicted with paths of other two groups (left part). Then path for G_1 has been successfully replanned (right part).

Standley uses ID in combination with the A* algorithm. While planning, it is preferred to find paths that create the least possible amount of conflicts with other groups that have already planned paths. For this purpose, the *conflict avoidance table* is created.

3.1 Integration of ID into the SAT-based Approach

We will now describe integration of a variant of independence detection into the SAT-based solver. The standard SAT-based approach called MDD-SAT (Surynek et al., 2016) has still considerable limitation when compared to existing search based techniques. MDD-SAT considers the entire MAPF instance as a whole which significantly limits the scalability of this method. With large instances and many agents, MDD-SAT will eventually encounter formula of prohibitive size. In all other optimal search-based solvers some variant of ID is used to further mitigate the size of the instance needed to be tackled at once.

The logical step is hence to integrate a variant of ID into the SAT-based approach. We decided to do that for MDD-SAT as it is currently the state-of-the-art SAT-based solver for MAPF.

The SAT-based approach however requires modification of the original ID since in the propositional formula it is not possible to express preference that individual paths of groups of agents should avoid occupied positions in the *conflict avoidance table*. In the yes/no SAT environment we either manage to avoid occupied positions or not while in the negative case there is no easy tool how to control the number of conflicts.

The SAT-based version of ID works in similar way to the original version of Standley but instead of resolving conflicts between a pair of conflicting groups G_1 and G_2 it resolves conflict of group G_1 with all other groups. If this attempt is successful, G_1 is independent on others and the process can continue with resolving conflicts between remaining groups (see Figure 2 where G_1 has been made independent).

If the attempt to resolve conflict between G_1 and G_2 by making G_1 independent fails, the same is tried for G_2 . If the attempt for G_2 fails too groups are merged. The pseudo-code is shown as Algorithm 1.

Algorithm 1: Independence detection in the **SAT-based framework**. Conflict aviodance is strictly required.

```
assign each agent to a group;
plan a path for each group
  G1,…,Gk by MDD-SAT;
fill conflict avoidance table;
while conflicting groups exist
    G1, G2 = conflicting groups;
    if G1, G2 not conflicted before
        replan G1 by MDD-SAT with
             illegal moves
             based on {G1,…,Gk}-G1;
        if failed to replan G1
            replan G2 by MDD-SAT with
                 illegal moves
                 based on {G1,…,Gk}-G2;
        endif
    endif
    if no alternative paths for G1, G2
        merge G1 and G2;
        plan a path for
        new group by MDD-SAT;
    endif
    update conflict avoidance table;
end
return combined paths of all groups;
```

If the attempt to resolve conflict between G_1 and G_2 by making G_1 independent fails, the same is tried for G_2 . If the attempt for G_2 fails too groups are merged. The pseudo-code is shown as Algorithm 1.

In contrast to original ID we strictly require avoidance with respect to the conflict avoidance table instead of stating it as a preference only. The SAT approach does not allow to express a preference like in the search based algorithm. This is the reason why ID in the SAT-based solver differs from the original one.

4. EXPERIMENTS

We performed experimental comparison of the suggested MDD-SAT+ID solver with other state-of-the-art solvers – namely with the previous best SAT-based solver MDD-SAT and also with state-of-the-art search-based algorithms ICTS (Sharon et al., 2013) and ICBS (Boyarski et al., 2015).

The MDD-SAT+ID has been implemented in C++ as an extension of an existing implementation of the MDD-SAT solver. We used Glucose 3.0 (Audemard, Simon, 2013) in MDD-SAT and MDD-SAT+ID which is a top performing SAT solver according to the recent SAT Competitions (Balint et al., 2015).

ICTS and ICBS have been implemented in C#. The original implementations of these algorithms have been used. All the tests were run on Xeon 2Ghz, and on Phenom II 3.6Ghz, both with 12 Gb of memory.

The experimental setup followed the scheme used in the literature (Silver, 2005, Sturtevant, 2012) which tests MAPF algorithms on 4-connected grids. Let us note however that all the suggested algorithms are designed and implemented for general undirected graphs (the fact that grids are used in the experiments is not exploited to increase efficiency of solving in any way).

We refer here a fragment of experiments that take place on small square grids of sizes 8×8, 16×16, and 32×32 with 10% of vertices occupied by obstacles. In this setup of the environment, we increased population of agents from 1 and observed the runtime of all the solvers until no solver was able to solve the instance within the given time limit of 300 seconds (this was 20 agents for 8×8 grid, and 40 and 60 for 16×16 and 32×32 girds respectively).

Ten randomly generated instances per number of agents were used. The initial positions were generated by choosing a subset of vertices randomly. The goal arrangement has been generated as a long random walk from the initial state following valid moves – this ensured solvability of all tested instances.

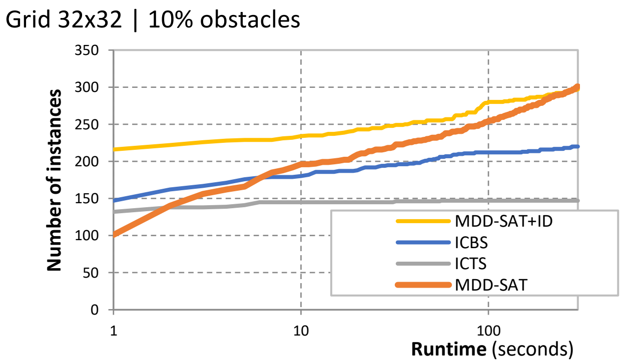

Figure 3: Results of experiments on grid map of sizes 32×32. Figure shows how many instances were solved within the given runtime. Clearly MDD-SAT and MDD-SAT+ID dominate in the test over search based algorithms ICTS and ICBS except few quickly solvable cases. Moreover, MDD-SAT+ID outperforms MDD-SAT in cases with low to medium density of agents.

The hypothesis is that the ID technique will be helpful in instances with medium density of agents. We also expect that in the case of low density of agents there will be some benefit of ID since many agents will just follow their shortest paths towards goals in such a case.

Furthermore, we expect rather negative effect of using ID in instances with high density of agents. This is because of the fact that most agents will be gradually merged into a large group while the process of merging represents an overhead in such a case.

A fragment of experimental results for the small grids is shown in Figure 3. MDD-SAT+ID clearly wins in low to medium density of agents. However, for the higher density, it tends to be outperformed by the original MDD-SAT.

5. CONCLUSION

It can be generally observed that ID brings worthwhile improvement to MDD-SAT solver which by itself performs very well. Experimental results indicate that there is a certain range of the density of agents though not precisely determined in our evaluation in which ID is beneficial while outside this range it cases an overhead.

The implementation of ID within the MDD-SAT+ID solver did not use any special reasoning about what groups of agents should be merged or not. The groups were processed in the ordering given by the original ordering of agents. We expect that more careful reasoning about merging can bring yet more improvements.

The suggested MDD-SAT+ID solver which is the result of integration of ID into an existing SAT-based MAPF solver MDD-SAT became a new state-of-the-art in optimal SAT-based MAPF solving. Moreover, the new MDD-SAT+ID performs well with respect to best search based solvers ICTS and ICBS though we cannot say there is a universal winner.

There are important future research directions which we just touched in this work:

(i) First, the experimental evaluation indicates the need to develop concepts for more precise classification of density and interaction among agents. Such a classification should ultimately lead to determining automatically in which cases ID would be beneficial and in which cases not.

(ii) The second future direction would become very apparent after a close look at the implementation. Currently we take groups of agents to be merged in the same order as they appear in the input. A more informed consideration which groups of agents should be merged may bring further reduction of the size of groups of agents.

6. ACKNOWLEDGEMENT

This paper is supported by a project commissioned by the New Energy and Industrial Technology Development Organization Japan (NEDO) and a joint grant of the Israel Ministry of Science and the Czech Ministry of Education Youth and Sports number 8G15027.

REFERENCES

[Audemard, G., Simon, L., 2013] The Glucose SAT Solver. http://labri.fr/perso/lsimon/glucose/, 2013.

[Audemard, G., Simon, L., 2009] Predicting Learnt Clauses Quality in Modern SAT Solvers. Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 399-404, IJCAI.

[Balint, A., Belov, A., Heule, M., Järvisalo, M., 2015] SAT 2015 competition. http://www.satcompetition.org/, 2015.

[Berg, J. van den, Snoeyink, J., Lin, M. C., Manocha, D., 2010] Centralized path planning for multiple robots: Optimal decoupling into sequential plans. Proceedings of Robotics: Science and Systems V, University of Washington, 2009, The MIT Press.

[Biere, A., Heule, M., van Maaren, H., Walsh, T., 2009] Handbook of Satisfiability. IOS Press.

[Boyarski, E., Felner, A., Stern, R., Sharon, G., Tolpin, D., Betzalel, O., Shimony, S., 2015] ICBS: Improved Conflict-Based Search Algorithm for Multi-Agent Pathfinding. Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015), pp. 740-746, IJCAI.

[Huang, R., Chen, Y., Zhang, W., 2010] A Novel Transition Based Encoding Scheme for Planning as Satisfiability. Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010), AAAI Press.

[Kautz, H., Selman, B., 1999] Unifying SAT-based and Graph-based Planning. Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 1999), pp. 318-325, Morgan Kaufmann.

[Kim, D., Hirayama, K., Park, G.-K, 2014] Collision Avoidance in Multiple-Ship Situations by Distributed Local Search. Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol-ume 18(5), pp. 839-848, Fujipress.

[Kornhauser, D., Miller, G. L., Spirakis, P. G, 1984] Coordinating Pebble Motion on Graphs, the Diameter of Permutation Groups, and Applications. Proceedings of the 25th Annual Symposium on Foundations of Com-puter Science (FOCS 1984), pp. 241-250, IEEE Press.

[Ryan, M. R. K., 2008] Exploiting Subgraph Structure in Multi-Robot Path Planning. Journal of Artificial Intel-ligence Research (JAIR), Volume 31, 2008, pp. 497-542, AAAI Press.

[Sharon, G., Stern, R., Goldenberg, M., Felner, A., 2013] The increasing cost tree search for optimal multi-agent pathfinding. Artificial Intelligence, Volume 195, pp. 470-495, Elsevier.

[Sharon, G., Stern, R., Felner, A., Sturtevant, N. R., 2015] Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence, 219, 40-66, Elsevier.

[Silver, D., 2005] Cooperative Pathfinding. Proceedings of the 1st Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2005), pp. 117-122, AAAI Press.

[Standley, T., 2010] Finding Optimal Solutions to Cooperative Pathfinding Problems. Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-2010), pp. 173-178, AAAI Press.

[Sturtevant, N. R., 2012] Benchmarks for Grid-Based Pathfinding. IEEE Transactions on Computational Intelligence and AI in Games, Volume 4(2), pp. 144-148, IEEE Press.

[Surynek, P., 2009] A Novel Approach to Path Planning for Multiple Robots in Biconnected Graphs. Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), pp. 3613-3619, IEEE Press.

[Surynek, P., 2014] Compact Representations of Cooperative Path-Finding as SAT Based on Matchings in Bi-partite Graphs. Proceedings of the 26th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2014), pp. 875-882, IEEE Computer Society.

[Surynek, P., Felner, A., Stern, R., Boyarski, E., 2016] Efficient SAT Approach to Multi-Agent Path Finding Under the Sum of Costs Objective. Proceedings of 22nd European Conference on Artificial Intelligence (ECAI 2016), pp. 810-818, IOS Press.

[Wang, K. C., Botea, A., 2008] Fast and memory-efficient multi-agent pathfinding, Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS 2008), pp. 380-387, AAAI Press.

[Wilson, R. M., 1974] Graph Puzzles, Homotopy, and the Alternating Group. Journal of Combinatorial Theory, Ser. B 16, pp. 86-96, Elsevier.