Propositional Satisfiability Benchmarks Constructed from Multi-Robot Path Planning on Graphs

Pavel Surynek

Abstract **— A propositional satisfiability (SAT) benchmark motivated by planning paths for multiple robots on graphs is described in this short paper. It is suggested to model the question if robots can find paths in a graph to given goal vertices in the given number of time steps as propositional satisfiability. The problem, its propositional model, and benchmark generator for grid environments are described.**

I. Where the Multi-Robotic Problem Comes From

Multi-robot path planning (MRPP, also referred *as cooperative path-finding – CPF*) on graphs [5], [6] is an abstraction for centralized navigation of multiple mobile robots (distinguishable but same in other aspects). Each robot has to relocate itself from a given initial location to a given goal location while it must not collide with other robots and obstacles. Plans as sequences of movements for each robot are constructed in advance by a centralized planner which can fully observe the situation.

The problem of navigating a group of mobile robots or other movable units has many practical applications. Except the classical case with mobile robots let us mention traffic optimization, relocation of containers [5], or movement planning of units in RTS computer games.

To be able to tackle the problem a graph-based abstraction is often adopted – the environment is represented as an undirected graph with at most one robot in a vertex. Edges can be traversed by robots.

We describe a MRPP problem formally and develop SAT encoding for it in the following sections. Then an instance generator for MRPP on 4-connected grids is described.

II. From Graph Formulation to SAT Encoding

Our encoding of MRPP will be introduced through finite domain integer programming. After creating integer model, the integer variables and constraints will be replaced with vectors of propositional variables (bit-vectors) and corresponding clauses.

A. Multi-robot Path Planning on Graphs (MRPP)

Let $G = (V, E)$ be an undirected graph and let $R = \{r_1, r, \ldots, r_n\}$ be a set of robots where $|R| < |V|$. The arrangement of robots in G will be described by a uniquely invertible function $\alpha: R \longrightarrow V$. The interpretation is that a robot $r \in R$ is located in a vertex $\alpha(r)$. A generalized inverse of α denoted as $\alpha^{-1}: V \longrightarrow R \cup \{\perp\}$ will provide us a robot located in a given vertex or $\perp$ if the vertex is empty.

An arrangement of robots at time step $i \in \mathbb{N}_0$ will be denoted as α_i. If we formally express rules on movements in terms of location function then we have following *transition constraints*:

- *(i)* $\forall r \in R$ either $\alpha_i(r) = \alpha_{i+1}(r)$ or $\{\alpha_i(r), \alpha_{i+1}(r)\} \in E$ holds (*robots move along edges or do not move at all*),
- (ii) $\forall r \in R\ \ \alpha_i(r) \neq \alpha_{i+1}(r) \Rightarrow \alpha_i^{-1}(r) = \perp$ (*robots move to empty vertices only*), and
- (iii) $\forall r, s \in R\ \ r \neq s \Rightarrow \alpha_{i+1}(r) \neq \alpha_{i+1}(s)$ (*no two robots enter the same target vertex*).

The initial arrangement is α_0 and α^+ will denote the goal arrangement. An instance of MRPP is then given as quadruple $[G, R, \alpha_0, \alpha^+]$. The task is to transform α_0 to α^+ so that transition constraints are preserved between all the consecutive time steps.

<u>*Definition 1 (solution, makespan).*</u> Let $\Sigma = [G, R, \alpha_0, \alpha^+]$ be an instance of CPF. A *solution* of Σ is a sequence of arrangements $\alpha_0, \alpha_1, \ldots, \alpha_\mu$ where $\alpha_\mu = \alpha^+$ and transition constraints are satisfied between α_{i-1} and α_i for every $i = 1, \ldots, \mu$. The number μ is called a *makespan* of the solution. The shortest possible makespan of Σ will be denoted as $\mu^*(\Sigma)$. □

It is known that finding $\mu^*(\Sigma)$ is NP-hard [4]. If makespan sub-optimal solution is sufficient then polynomial time solving techniques from [3] can be used. An example of MRPP instance on a graph represented by a 4-connected grid is shown in Figure 1.

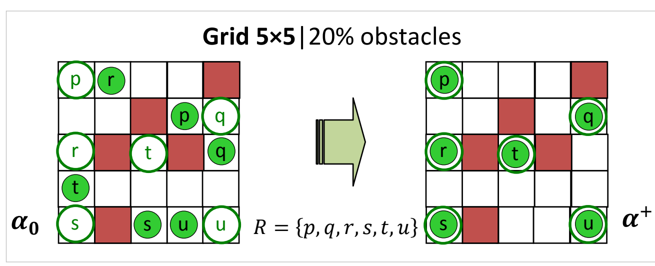

Figure 1. A typical **random MRPP instance on a grid of size 5×5** with 20% of positions occupied by obstacles.

B. k-Level MRPP Encoding as Integer Programming

An incomplete approach from domain independent planners SASE [1] and SATPlan [2] can be adopted to find makespan optimal solutions of MRPP. A question whether there exists a solution of the given MRPP of makespan k is modeled as propositional satisfiability. A solution of the optimal makespan can be found by trying larger and larger makespans in a case the MRPP instance is solvable (the unsolvability cannot be detected by this approach).

Unlike domain independent planners SASE and SATPlan we use a propositional encoding specially designed for

Pavel Surynek is with Charles University in Prague, Faculty of Mathematics and Physics, Department of Theoretical Computer Science and Mathematical Logic, Malostranské náměstí 25, 118 00 Praha 1, Czech Republic (phone: +420 221 914 245; fax: +420 221 914 323; e-mail: pavel.surynek@mff.cuni.cz).

Supported by Charles University in Prague within the PRVOUK project and by Czech Science Foundation under the contract number GAP103/10/1287.

MRPP. The employed encoding called *inverse* has been developed in [7] and is significantly smaller in terms of the number of variables and clauses than SASE and SATPlan encodings on the same MRPP instances.

Basically we need to model arrangements of robots at individual time steps and introduce transition constraints into the model. In the inverse encoding, the arrangement of robots at time step i is modeled by state variables $\mathcal{A}_i^v$ for $v \in V$ that represent inverse location function at the time step i. Next, there are state variables $\mathcal{T}_i^v$ for $v \in V$ that represent actions taken in vertices at time step i. An *outgoing action* into some of vertex neighbors or an *incoming action* from some of vertex neighbors or *noop* can be taken in each vertex. The domain of $\mathcal{T}_i^v$ consists of $2\,\deg_G(v) + 1$ values to represent all the possible actions. It is necessary to introduce some ordering on neighbors of each vertex to be able to assign concrete actions to elements of the domain of $\mathcal{T}_i^v$. Suppose that we have a function $\sigma_v: \{u | \{v, u\} \in E\} \longrightarrow \{1,2, ..., \deg_G(v)\}$ and its inverse σ_v^{-1} that implements this ordering of neighbors.

Definition 2 (inverse encoding). The i-th level of *inverse encoding* consists of the following integer interval **state variables**:

- $\mathcal{A}_i^v \in \{0,1,2, \dots, n\}$ for all $v \in V$ such that
  $\mathcal{A}_i^v = j$ iff $\alpha_i(r_j) = v$
- $\mathcal{T}_i^v \in \{0,1,2, \dots, 2\deg_G(v)\}$ for all $v \in V$ such that
  $\mathcal{T}_i^v = 0$ iff no-op was selected in v;
  $\mathcal{T}_i^v = \sigma_v(u)$ iff an outgoing primitive action with the target $u \in V$ was selected in v;
  $\mathcal{T}_i^v = \deg_G(v) + \sigma_v(u)$ iff an incoming primitive action with $u \in V$ as the source was selected in v.

and **constraints**:

- $\mathcal{T}_i^v = 0 \Rightarrow \mathcal{A}_{i+1}^v = \mathcal{A}_i^v$ for all $v \in V$ (**no-op** case);
- $0 < \mathcal{T}_i^v \leq \deg_G(v) \;\Rightarrow\; \mathcal{A}_i^u = 0 \wedge \mathcal{A}_{i+1}^u = \mathcal{A}_i^v \wedge \mathcal{T}_i^u = \sigma_u(v) + \deg_G(u)$ where $u = o_v^{-1}(\mathcal{T}_i^v)$ for all $v \in V$ (**outgoing** robot case);
- $\deg_G(v) < T_i^v \leq 2 * \deg_G(v) \;\Rightarrow\; \mathcal{T}_i^u = \sigma_u(v)$ where $u = \sigma_v^{-1}(\mathcal{T}_i^v - \deg_G(v))$ for all $v \in V$ (**incoming** robot case). □

C. Translation of IP Model of MRPP to SAT

The encoding is built upon integer finite domains variables. We eventually need propositional encoding which is obtained by translating integer state variables into bit vectors. If the state variable has N states (N elements in its domain) then we need $\lceil \log_2 N \rceil$ propositional variables to represent it.

If we are asking whether there is a solution of makespan k we need to build k levels. The initial arrangement α_0 is encoded in $\mathcal{A}_0^v$. Analogically $\mathcal{A}_k^v$ are set to the goal arrangement α^+.

III. MRPP on Grids Instance Generator

A classical MRPP benchmark introduced in [6] takes place on a 4-connected grid of certain size into which obstacles are placed randomly by excluding randomly selected nodes. Initial and goal positions for robots are random as well. In all the cases random selection is uniform from the set of remaining items. Our instance generator produces SAT encodings for these benchmarks. Several parameters are accepted by the generator:

- *size* of the grid – dimensions *height* × *width*
- *probability* of obstacles – placed randomly/uniformly
- number of *robots* – placed randomly/uniformly
- number of *levels* – corresponds to the *makespan*
- random *seed*

A. Simple Knowledge Compilation into the SAT Encoding

A simple knowledge compilation into the presented encoding is done by our instance generator. It is checked if a given robot can occur in a given vertex at a given time step. Such occurrence of a robot excludes existence of a solution if the vertex cannot be reached from the initial position in the given number of time steps or if the goal position cannot be reached in the remaining number of time steps along shortest paths.

B. Properties, Parameters and Difficulty

A property having the most significant impact on the difficulty of MRPP solving is the **intensity of interactions** among robots during their movement. It is more difficult to solve a problem when robots need to intensively avoid each other regardless of the solving method applied [7], [8]. Intensity of interaction is directly changed by the *size* of the grid, probability of *obstacles*, and the number of *robots*.

Notice also that the SAT model encodes bounded MRPP by certain number of levels. The most difficult cases appear for the number of *levels* around the optimal makespan [2]. On the other hand instances with few levels can be quickly identified as unsolvable. However, it is typically more difficult to discover solvability of instances with many levels due to increasing size of the instance.

Discussion and Future Work

Several other encodings of MRPP were investigated by the author. The presented *inverse* encoding is the most compact one if the number of robots is relatively high.

There is still room for improving encodings by compiling more sophisticated knowledge into it. Further compacting the encoding at the bit level is also planned.

References

[1] R. Huang, Y. Chen, W. Zhang. "A Novel Transition Based Encoding Scheme for Planning as Satisfiability," Proceedings of AAAI 2010, AAAI Press, 2010.

[2] H. Kautz, B. Selman, "Unifying SAT-based and Graph-based Planning," Proceedings of IJCAI 1999, pp. 318-325, Morgan Kaufmann, 1999.

[3] D. Kornhauser, G. L. Miller, P. G. Spirakis, "Coordinating Pebble Motion on Graphs, the Diameter of Permutation Groups, and Applications," Proceedings of FOCS 1984, pp. 241-250, IEEE Press, 1984.

[4] D. Ratner, M. K. Warmuth, "Finding a Shortest Solution for the N × N Extension of the 15-PUZZLE Is Intractable." Proceedings of AAAI 1986, pp. 168-172, Morgan Kaufmann, 1986.

[5] M. R. K. Ryan, "Exploiting Subgraph Structure in Multi-Robot Path Planning," JAIR, Volume 31, 2008, pp. 497-542, AAAI Press, 2008.

[6] D. Silver, "Cooperative Pathfinding," Proceedings of AIIDE 2005, pp. 117-122, AAAI Press, 2005.

[7] P. Surynek, "Towards Optimal Cooperative Path Planning in Hard Setups through Satisfiability Solving," Proceedings of PRICAI 2012, pp. 564-576, LNCS 7458, Springer, 2012.

[8] T. S. Standley, "Finding Optimal Solutions to Cooperative Pathfinding Problems," Proceedings of AAAI 2010, AAAI Press, 2010.