# Optimal Cooperative Path-Finding with Generalized Goals in Difficult Cases

**Pavel Surynek**

Charles University in Prague
Faculty of Mathematics and Physics, Malostranské náměstí 25, 118 00 Praha 1, Czech Republic

pavel.surynek@mff.cuni.cz

**Abstract**

We suggest to employ propositional satisfiability techniques in solving a problem of cooperative multi-robot path-finding optimally. Several propositional encodings of path-finding problems have been suggested recently. In this paper we evaluate how efficient these encodings are in solving certain cases of cooperative path-findings problems optimally. Particularly, a case where robots have multiple optional locations as their targets is considered in this paper.

## Introduction and Context

*Multi-robot path planning* (MRPP, also referred *as cooperative path-finding – CPF*) on graphs (Silver, 2005; Ryan, 2006; Wang & Botea, 2011; Luna & Berkis, 2011) is an abstraction for centralized navigation of multiple mobile robots (distinguishable but all the 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 (Ryan, 2006), 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 are used to model passable regions.

We describe a generalization of MRPP (gMRPP). In the classical MRPP each robot has a single vertex as its goal. This is however impractical in many situations – sometimes we need robots that are not distinguished with respect to their goals. To overcome this limitation we introduce multiple optional goals for robots, that is the target for a robot is no longer a single vertex but an non-empty set of vertices (a robot should have at least one target vertex not to compromise solvability directly at the beginning).

Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

## gMRPP: MRPP with Generalized Goals

Let $G = (V, E)$ be an undirected graph and let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of robots where $|R| < |V|$. The arrangement of robots in $G$ will be described by a uniquely invertible function $\alpha: R \rightarrow V$. The interpretation is that a robot $r \in R$ is located in a vertex $\alpha(r)$. A generalized inverse of $\alpha$ denoted as $\alpha^{-1}: V \rightarrow R \cup \{\perp\}$ 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 $\alpha_i$. If we formally express rules on movements in terms of location function then we have the 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) $\forall r \in R \;\; \alpha_i(r) \neq \alpha_{i+1}(r) \Rightarrow \alpha_i^{-1}(\alpha_{i+1}(r)) = \perp$
(*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 $\alpha_0$. Each robot has at least one vertex as its target.

A set of goal vertices to robots will be assigned by a function $A^+: R \rightarrow 2^V$ such that $|A^+(r)| \geq 1 \;\; \forall r \in R$. An instance of gMRPP is then given as quadruple $[G, R, \alpha_0, A^+]$.

We say that $A^+$ is satisfied by an arrangement of robots $\alpha$ if and only if $\forall r \in R \;\; \alpha(r) \in A^+(r)$. The task in gMRPP is to transform $\alpha_0$ into an arrangement $\alpha^+$ so that transition constraints are preserved between all the consecutive time steps and $A^+$ is satisfied by $\alpha^+$.

*Definition 1 (solution, makespan).* Let $\Gamma = [G, R, \alpha_0, A^+]$ be an instance of gMRPP. A *solution* of $\Gamma$ is a sequence of arrangements of robots $\alpha_0, \alpha_1, \ldots, \alpha_\mu$ where $\alpha_\mu$ satisfies $A^+$ and transition constraints are satisfied between $\alpha_{i-1}$ and $\alpha_i$ for every $i = 1, \ldots, \mu$. The number $\mu$ is called a *makespan* of the solution. □

An example of gMRPP instance on a graph represented by a 4-connected grid is shown in Figure 1.

Having a solution of the shortest possible makespan we say the solution to be *makespan optimal* or shortly *optimal*. We are interested in generating optimal solutions to gMRPP in this paper.

It is known that finding an optimal solution is NP-hard (Ratner & Warmuth, 1986). The result has been shown for MRPP where robots have single targets. Thus there is little chance to find optimal solutions of gMRPP efficiently in the general case.

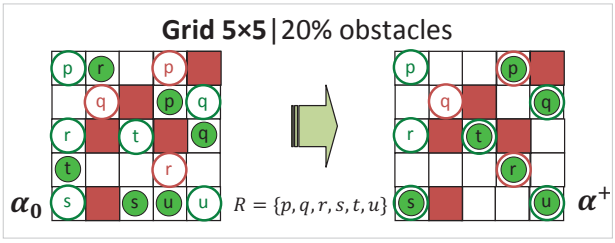


Figure 1. A typical **random gMRPP instance on a grid of size 5×5** with 20% of positions occupied by obstacles. Robots $p$, $q$, and $r$ has two optional goals.

Since the goal condition for a single robot arranged by $\alpha$ can be naturally expressed as formula in the disjunctive form (exactly a robot $r$ is in its destination if and only if $\bigvee_{v \in A^+(r)} v = \alpha(r)$) we consider that it is suitable to model gMRPP as propositional satisfiability.

An incomplete approach from domain independent planners like SASE (Huang et al., 2010) and SATPlan (Kautz & Selman, 1999) can be adopted to find makespan optimal solutions of gMRPP. A question whether there exists a solution of the given gMRPP 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 propositional encodings specially designed for gMRPP. Several special encodings for MRPP have been suggested in (Surynek, 2012a, 2012b).

All the special encodings are significantly smaller in terms of the number of variables and clauses than domain-independent SASE and SATPlan encodings on the same MRPP instances. Also special encodings are solved faster by a SAT solver (Surynek, 2012a, 2012b).

## Adapting Encodings for gMRPP

The base encodings of our choice are called *inverse* and *all-different* in (Surynek, 2012a). We show how the inverse encoding should be adapted; adapting the all-different encoding is analogical and is omitted.

The arrangement of robots at all the considerable time steps is represented by the generalized inverse $\alpha^{-1}$ in the inverse encoding. That is, we know which robot is located in each vertex (and whether the vertex is empty or not). In case of the all-different encoding, the arrangement is represented by $\alpha$ (we know which robot is located in a vertex).

To explain the encoding in a more natural way a finite domain integer abstraction is used first. The propositional encoding is subsequently obtained by replacing integer variables with vectors of propositional variables (bit-vectors) and integer constraints with their propositional equivalents. Notice that additional propositional variables may be added to allow hierarchical translation (Tseitin, 1968). Using bit-vectors to represent state variables is a key technique to make the encoding compact (Rintanen, 2006).

There are visible variables $\mathcal{A}_i^v \in \{0,1,2,\ldots,n\}$ in the inverse encoding to represent inverse locations of robots. That is, $\mathcal{A}_i^v = j$ if and only if $\alpha_i(r_j) = v$ (robot $r_j$ is located in $v$ at time step $i$, 0 is reserved for an empty vertex). Additional variables are used to model actions and to establish valid transitions between consecutive time steps – for details see (Surynek, 2012a).

The only change we need to make in the inverse encoding to capture gMRPP is that instead of the original goal constraints we should use the following one. Suppose that the encoding is build for $k$ levels:

$$\mathcal{A}_k^v = 0 \vee \bigvee_{j \mid v \in A^+(r_j)} \mathcal{A}_k^v = j$$

### Knowledge Compilation

It is relatively easy to compile various types of abstract knowledge into the propositional encoding. A certain kind of mutex and reachability heuristic compilation into the encoding of MRPP is studied in (Surynek, 2013). Here we compile results of reachability heuristic into the encoding (mutex reasoning is not used here). Again suppose $k$ levels of the encoding. The following constraints can be added for every vertex $v \in V$ and robot $r_j \in R$ at time step $i$:

$$\left[ \mathrm{dist}_G(\alpha_0(r_j), v) > i \vee \bigwedge_{u \in A^+(r_j)} \mathrm{dist}_G(v, u) > k - i \right] \Rightarrow \mathcal{A}_i^v \neq j$$

In other words, robot $r_j$ cannot stay in $v$ at time step $i$ if it cannot reach $v$ in $i$ steps or if it cannot reach any of its goals in the remaining number of time steps.

Such knowledge compilation can significantly reduce the search space. Notice that the presented reachability heuristic is relatively simple knowledge at the abstract level of gMRPP however it may be hard to discover at the propositional level.

## Experimental Evaluation

We made an experimental evaluation of the performance of SAT-based solving of gMRPP instances. Here we present

part of results regarding the difficulty of solving depending on the number of alternative goal vertices of robots. Testing gMRPP instances consist of a 4-connected grid of the size 6×6. Robots are initially placed randomly. Each robot has at least one target vertex. The total number of target vertices was selected randomly (uniformly) between 1 and the given upper bound while the upper bound was changed in the experiments – we tried 1, 2, 3, and 4.

All the source code and experimental data to allow reproducibility of presented experiments are available on: http://ktiml.mff.cuni.cz/~surynek/research/sara2013.

## Runtime Comparison

The runtime comparison[1] is shown in Figure 2. and Figure 3. We used `MiniSat 2.2` (Eén & Sörensson, 2004) to solve the encoded *k*-level bounded instances of gMRPP. The number of robots varied from 1 to 15; 10 random instances were solved for each number of robots. Average values are presented.

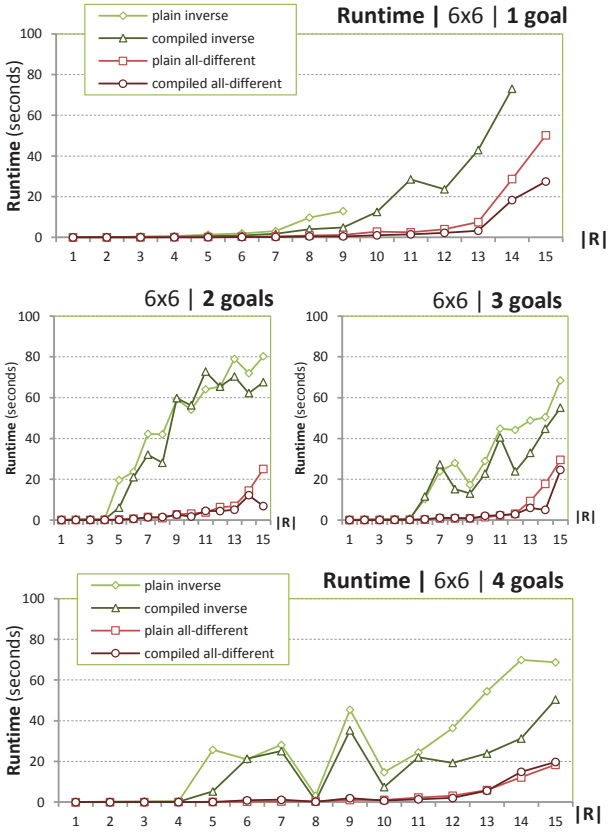


Figure 2. **Runtime comparison** 4 encodings on instances with different number of optional goals. Each robot has random number of goals choosen uniformly between 1 and the upper bound which is 1, 2, 3, or 4. The graph is 4-connected grid of size 6×6 with 10% vertices occupied with obstacles.

[1] All the tests were run on Intel Xeon 2.0 GHz with 12 Gb of RAM under Ubuntu Linux 10.04.

Two basic propositional encodings are tested – an *inverse* one where visible variables represent $\alpha^{-1}$ and an *all-different* one where visible variables represent $\alpha$ (occupancy of vertices by at most one robot is expressed as the all-different constraint). Further details on these encodings are omitted here – we refer the reader to (Surynek, 2012a, 2012b). A plain variant and a variant with compiled reachability heuristic was tried for both encodings.

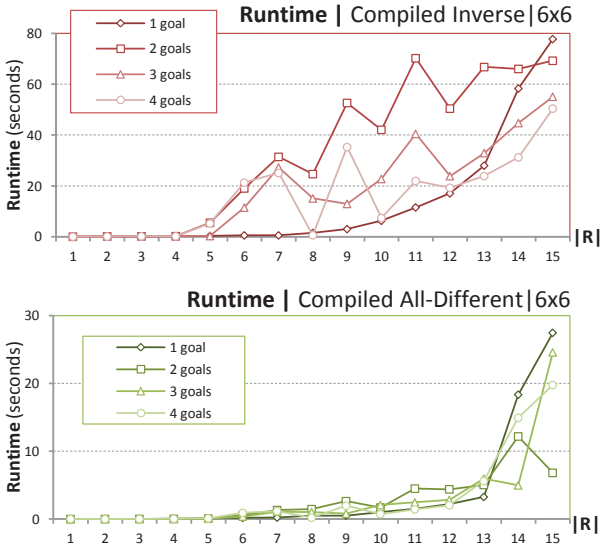


Figure 3. **Runtime development** with the increasing number of goals. Instances with one goal seems to be easy with few robots but become quicly hard as the number of robots increases. The problem seems to relax by adding goals in other cases.

It can be observed that allowing more alternative goal vertices relaxes the problem – it is easier to solve. The most difficult cases appear when there are one or two goals for each robot. Notice that if all the robots have uniformly one goal the instance is easy for few robots however it is getting increasingly difficult as the number of robots increases.

The benefit of knowledge compilation can be also observed. It appears to be more beneficial for the inverse encoding.

Both tested variants of the all-different encoding outperformed the inverse one. Let us note that the performance of the all-different encoding quickly degrades with increasing number of robots and it is eventually outperformed by the inverse encoding in cases with many robots (Surynek, 2012b).

## Makespan Evaluation

Average optimal makespans for tested gMRPP instances are shown in Figure 4. Again, with the increasing number of goal vertices the optimal makespan tends to be lower (robots can chose a goal vertex closer to their initial location). The average optimal makespan is calculated from 10 results for each number of robots in the graph.

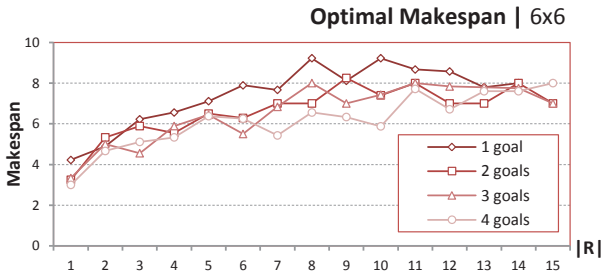


Figure 4. **Optimal makespans** for instances from Figure 2. Notice that more optional goal relaxes the problem – robots can chose goal vertex closer to their initial location.

## Conclusion and Future Work

The multi-robot path-planning (gMRPP) with generalized goals has been introduced in this paper. The generalization consists in allowing robots to have more than a single vertex as its targets. We suggested to solve the problem optimally via modeling it as propositional satisfiability. Initial experimental results targeted evaluating how the number of alternative goals affects the difficulty of solving are presented.

More extensive evaluation with larger instances and more scenarios is planned. We also plan to compare optimal SAT-based solving with other optimal solving methods for MRPP such as (Standley & Korf, 2011).

It seems that compiling knowledge discovered in the problem at the abstract level into propositional encodings represents a powerful technique. We would like to further investigate this aspect. For example solution improvements (Wang et al., 2011) seems to be easily adaptable for compilation in this sense. An interesting knowledge that may be hard to discover by a SAT solver but easily accessible at the abstract level is represented by *duality* in permutation groups (Felner et al., 2005).

## Acknowledgement

This work is supported by the Charles University in Prague within the UNCE and the PRVOUK (section number P46) projects and by the Czech Science Foundation under the contract number GAP103/10/1287. I would like thank reviewers for their suggestions and corrections.

## References

Eén, N., Sörensson, N., 2004. *An Extensible SAT-solver.* Proceedings of Theory and Applications of Satisfiability Testing (SAT 2003), pp. 502-518, LNCS 2919, Springer.

Felner, A., Zahavi, U., Schaeffer, J., Holte, R. C., 2005. *Dual Lookups in Pattern Databases.* Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 103-108, Professional Book Center.

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.

Luna, R., Berkis, K., E., 2011. *Push-and-Swap: Fast Cooperative Path-Finding with Completeness Guarantees.* Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 294-300, IJCAI/AAAI Press.

Ratner, D., Warmuth, M. K.,1986. *Finding a Shortest Solution for the N × N Extension of the 15-PUZZLE Is Intractable.* Proceedings of the 5th National Conference on Artificial Intelligence (AAAI 1986), pp. 168-172, Morgan Kaufmann.

Rintanen, J., 2006. *Compact Representation of Sets of Binary Constraints.* Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006), pp. 143-147, IOS Press 2006.

Ryan, M. R. K., 2008. *Exploiting Subgraph Structure in Multi-Robot Path Planning.* Journal of Artificial Intelligence Research (JAIR), Volume 31, pp. 497-542, AAA Press.

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. S., Korf, R. E., 2011. *Complete Algorithms for Cooperative Pathfinding Problems.* Proceedings of IJCAI 2011, 668-673, IJCAI/AAAI Press.

Surynek, P., 2012a. *Towards Optimal Cooperative Path Planning in Hard Setups through Satisfiability Solving.* Proceedings of the 12th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2012), pp. 564-576, Lecture Notes in Computer Science 7458, Springer.

Surynek, P., 2012b. *On Propositional Encodings of Cooperative Path-finding.* Proceedings of the 24th International Conference on Tools with Artificial Intelligence (ICTAI 2012), pp. 524-531, IEEE Press.

Surynek, P., 2013. *Mutex Reasoning in Cooperative Path Finding Modeled as Propositional Satisfiability.* Submitted to 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Tseitin, G., 1968. *On the complexity of derivation in propositional calculus.* Studies in Constructive Mathematics and Mathematical Logic, pp. 115–125, 1968. Reprinted in Automation of Reasoning, Volume 2, pp. 466–483, Springer.

Wang, K. C., Botea, A., Kilby, P., 2011. *Solution Quality Improvements for Massively Multi-Agent Pathfinding.* Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI 2011), AAAI Press.

Wang, K. C., Botea, A., 2011. *MAPP: a Scalable Multi-Robot Path Planning Algorithm with Tractability and Completeness Guarantees.* JAIR, Volume 42, pp. 55-90, AAAI Press.