Guaranteed Plans for Multi-Robot Systems via Optimization Modulo Theories*

Francesco Leofante
RWTH Aachen University and University of Genoa
leofante@cs.rwth-aachen.de

Introduction

Industries are on the brink of widely accepting a new paradigm for organizing production by having autonomous robots manage in-factory processes. This transition from static process chains towards more automation and autonomy poses new challenges in terms of, *e.g.*, efficiency of production processes.

The RoboCup Logistics League (RCLL) (Niemueller, Lakemeyer, and Ferrein 2015) has been proposed as a realistic testbed to study the above mentioned problem at a manageable scale. In RCLL, teams of robots manage and optimize the material flow according to dynamic orders in a simplified factory environment. In particular, robots have to transport workpieces among several machines scattered around the factory shop floor. Each machine performs a specific processing step, orders that denote the products which must be assembled with these operations are posted at runtime and require quick planning and scheduling. Orders also come with a delivery time window, therefore introducing a temporal component into the problem.

Though there exist successful heuristic approaches to solve the underlying planning and scheduling problems, a disadvantage of these methods is that they provide no guarantees about the quality of the solution, as observed in (Bensalem, Havelund, and Orlandini 2014). A promising solution to this problem is offered by the recently emerging field of Optimization Modulo Theories (OMT), where Satisfiability Modulo Theories (SMT) solving is extended with optimization functionalities – see, *e.g.*, (Sebastiani and Tomasi 2015).

Proposed Methodology

In my thesis, I propose to employ OMT solving to synthesize plans with optimality guarantees for multi-robot systems, using the RCLL as testbed. The approach I propose to use is based on symbolic reachability techniques used to solve the SMT-based bounded model-checking problem, which I extend to OMT – see (Biere et al. 1999) for the original SAT-based formulation.

Given the RCLL domain, we encode the *state space* using an indexed set of real-valued variables $x = \{x_1, \ldots, x_n\}$; we also use the vector notation $x = (x_1, \ldots, x_n)$. A *state* $s{=}(v_1, \ldots, v_n) \in \mathbb{R}^n$ specifies a real value $v_i \in \mathbb{R}$ to each state variable $x_i \in x$. The transition system underlying the RCLL domain is represented symbolically using Boolean combinations of linear constraints defining the *initial states* $I(x)$, the *transition relation* $T(x, x')$ and a set of *final* states $F(x)$, where $x'{=}\{x'_1, \ldots, x'_n\}$ is an indexed set of *next state* variables. As customary, the *bounded symbolic reachability problem* is a tuple *BSR*$=(I, T, F, p)$ and consists in finding a *path* π of length p – *i.e.*, a sequence of states $(s_0, \ldots, s_p)$ – such that the following holds:

$$\pi \vDash I(s_0) \wedge \left(\bigwedge_{0 \leq i < p} T(s_i, s_{i+1}) \right) \wedge \left(\bigvee_{0 \leq i \leq p} F(s_i) \right) \quad (1)$$

To extend bounded model-checking with optimization, we need to specify an indexed set of *reward* variables $r{=}\{r_1, \ldots, r_p\}$, where r_i is used to encode a reward for the transition from s_i to s_{i+1}. The total reward r_{tot} associated to a path $\pi = (s_0, \ldots, s_p)$ can be computed as:

$$\sum_{0 \leq i < p} r_i \quad (2)$$

An *optimal* bounded symbolic reachability problem can then be defined as a tuple *OBSR*$=(I, T, F, r_{tot}, p)$, whose solution is a sequence of states minimizing r_{tot} under the side condition that Eq. (1) holds.

Solvers such as `Z3` (de Moura and Bjørner 2008) and `OptiMathSAT` (Sebastiani and Trentin 2015) can be leveraged to reason on the model defined by Eqs. (1) and (2) and generate optimal plans that achieve desired high-level tasks.

However, when plans are executed on concrete systems, several modeling assumptions may be challenged, and the real applicability of computed solutions may be jeopardized. To tackle this problem, I propose to integrate OMT-based plan synthesis into an online execution and monitoring system based on `CLIPS`, a rule-based production system using forward chaining inference (Wygant 1989).

My long-term goal is to devise a system that unites the power of Optimization Modulo Theories with the flexibility of an on-line execution and monitoring system, to provide optimal solutions for high-level task planning, and runtime feedback on their feasibility.

---

[*] Joint work with E. Ábrahám, G. Lakemeyer, T. Niemueller and A. Tacchella.

Current Results

The example domain chosen for evaluating the proposed approach is based on the Planning Competition for Logistics Robots in Simulation (Niemueller et al. 2016).

A game in the competition is structured in two phases: (i) exploration and, (ii) production. I started with the exploration phase, during which two teams of robots must roam the environment and determine where the team's own machines are positioned. For this, the playing field is divided into twenty-four virtual zones, twelve of which belong to each of the two teams. However, only six of these zones contain machines. Therefore, the task is to efficiently assign three robots to twelve zones to identify the zones which contain a machine. Robots are given twenty minutes to perform exploration (planning time included), thus imposing tight timing constraints on our OMT-based module.

Although the problem formulation is simple, computing an optimal solution is highly challenging. This is due the fact that exploration is a variant of the multiple Traveling Salesman Problem, which is known to be NP-hard.

Our approach is presented in (Leofante et al. 2017), where by rigorous experimental analysis we show that naive encodings fail to cope with the complexity of the domain. After a thorough analysis, we identified ways to reduce solving time and eventually managed to decrease plan synthesis time from $5m$ to an average of $50s$. As a result, we also presented some general observations for the development of OMT-based approaches.

A prototypical implementation of our approach and its integration with the `CLIPS` online execution system is presented in (Niemueller et al. 2017). In our architecture, `CLIPS` controls the overall execution. When the game enters the exploration phase, `CLIPS` triggers the OMT solving process to synthesize a plan and encodes the world model in a message. The OMT solving module uses this knowledge to construct a first order formula over the reals and starts solving. If a plan can be found, the OMT plug-in sends it to the `CLIPS` executive. This plan is then synchronized with all robots, which execute their respective partial plans by invoking the appropriate basic behaviors through the behavioral and functional components of Fawkes[1].

Planned Contributions and Future Research

Building on the results obtained for the exploration phase, I started considering the second phase of the RCLL competition, *i.e.*, production. This part of the game poses challenges to the OMT solver that are different in nature with respect to the ones met when encoding the exploration phase.

One the one hand, production tasks are more constrained and therefore present less symmetries than exploration. On the other hand, they require more sophisticate robot-robot and robot-environment interactions, which affect both plan synthesis and execution.

To solve the production problem, we build formal models for robot motions, production processes, and for order schedules, deadlines and rewards. Again, we encode our model as a first order formula over the reals and ask OMT solvers to synthesize a plan. A submission describing the approach I propose is planned to the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'18).

In another work, I present the recent achievements obtained regarding the integration of our approach into `CLIPS`. A paper describing how our architecture manages the integration between plan synthesis and online execution and monitoring is planned.

To conclude, I would like to give an idea of how I plan to extend this research. First of all, it would be interesting to test the robustness of our approach. Since our current results deal with simulated environments only, I would like to run experiments using the real robotic setup used in RCLL. This will be done in collaboration with Niemueller and Lakemeyer, members of the current RCLL world-champion team. Empirical evidence shows that planners achieve impressive performances when it comes to fast exploration of large state-spaces. Starting from this observation, I would like to investigate whether planning heuristics can be imported as tactics into OMT solvers to speed up search.

References

Bensalem, S.; Havelund, K.; and Orlandini, A. 2014. Verification and validation meet planning and scheduling. *STTT* 16(1):1–12.

Biere, A.; Cimatti, A.; Clarke, E. M.; and Zhu, Y. 1999. Symbolic model checking without bdds. In *In Proc. of TACAS'99*, 193–207.

de Moura, L. M., and Bjørner, N. 2008. Z3: An efficient SMT solver. In *Proc. of TACAS'08*, 337–340.

Leofante, F.; Ábrahám, E.; Niemueller, T.; Lakemeyer, G.; and Tacchella, A. 2017. On the synthesis of guaranteed-quality plans for robot fleets in logistics scenarios via Optimization Modulo Theory. *CoRR* abs/1711.04259.

Niemueller, T.; Karpas, E.; Vaquero, T.; and Timmons, E. 2016. Planning competition for logistics robots in simulation. In *Proc. of PlanRob@ICAPS'16*.

Niemueller, T.; Lakemeyer, G.; Leofante, F.; and Ábrahám, E. 2017. Towards CLIPS-based task execution and monitoring with SMT-based decision optimization. In *Proc. of PlanRob@ICAPS'17*.

Niemueller, T.; Lakemeyer, G.; and Ferrein, A. 2015. The RoboCup Logistics League as a benchmark for planning in robotics. In *Proc. of PlanRob@ICAPS'15*.

Sebastiani, R., and Tomasi, S. 2015. Optimization modulo theories with linear rational costs. *ACM Trans. Comput. Log.* 16(2):12:1–12:43.

Sebastiani, R., and Trentin, P. 2015. OptiMathSAT: A tool for optimization modulo theories. In *Proc. of CAV'15*, 447–454.

Wygant, R. M. 1989. CLIPS: A powerful development and delivery expert system tool. *Computers & Industrial Engineering* 17(1–4).

---

[1] Fawkes is a component-based software framework for robotic real-time applications. URL: www.fawkesrobotics.org