The RoboCup Logistics League from a Planning Perspective

Marco De Bortoli and **Gerald Steinbauer**
Graz University of Technology

Abstract

Industry 4.0 raises the need for complex autonomous system. Smart factories, which make use of flexible and autonomous agents to organize the production chains, require some efforts for the integration of many topics from Robotics and Artificial Intelligence (AI). For this reason, the RoboCup Logistic League (RCLL) was created as a Flexible Production testbed for researchers working on these fields and to give the opportunity to different scientific communities to foster integration between their subjects. Among the others, planning and scheduling is a branch of AI which is crucial for the organization and execution of the workflow in smart production and logistics environment. In this paper the RCLL is investigated from this perspective to understand if and how it drives advanced planning approaches in production scenarios. Based on that evaluation a set of proposals is presented to foster discussion on how the competition can be improved for a better exchange with the planning community.

Introduction

In the last years we have seen a huge progess of all kind of automation in industry, stimulated by the progresses achieved by research in various fields.
The challenges posted by robotics are many and a number of different research fields are relevant. Depending of the actual environment, a single autonomous system may be concerned with many areas like localization, mapping, object manipulation and grasping, motion planning and, more in general, with planning and scheduling. The demand in the real world for complex autonomous systems able to tackle difficult problems foster the need for more integration and cooperation between different fields of research. The shift of production organizations from static process chains towards more automation is referred to as Industry 4.0 (Henning 2013) or Flexible Production. The increasing demand for flexibility, in contrast with common rigidity of automation, asks for new concepts and opens new interesting and challenging research questions.
In this perspective, the RoboCup Logistics League (RCLL) (Coelen et al.) was re-thought in 2015 with this integration in mind, is pursuing the goal of fostering closer cooperation between the planning community, used to work with very abstract environment, and the robotics community, highlighting the challenges of bringing an abstract decision into execution. This integration is not trivial, considering the issues coming from a real environment like uncertainty, high granularity of time and space, and unexpected events. In this competition, two fleets of three robots each compete in a logistic environment in assembling products with different configurations and complexities. The robots are mobile, the requested orders are given incrementally during the competition, and the robots must move between different stations in order to get the pieces they need. During the game, no human intervention is allowed.
Many fields are touched by the RCLL, making it a good testbed for innovative production concepts. Localization and mapping, grasping and planning, together with the integration between them, are the main aspects involved in this online multi-agent competition. Since the domain is driven by incremental orders and affected by failures, the team must be able to react to unexpected events and the insertion of new requests, in order to process them optimally. Moreover, there are different product configurations that need to be considered and features different complexity and rewards.

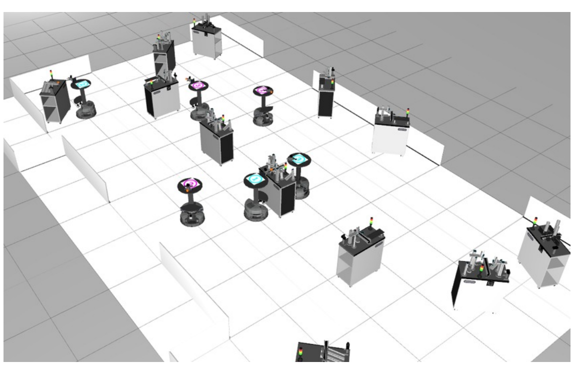
Althoug RCLL was designed to connect the planning and robotics community it is still not completely clear if the actual RCLL setting really stimulate the investigation and application of advanced planning methods for dynamic robotics domains. The contribution of the paper is therefore to foster a discussion regarding the involvement of advanced planning in the competition, in order to narrow the gap between between abstract classical planning and real scenarios in robotics. The paper is structured as following: in the next section, the actual setting of RCLL is briefly described. In following section investigates strategies adopted by the most successful teams attending the competition, like the Carologistics team from RWTH Aachen University and GRIPS team from Graz University of Technology. This investigation will give an insight into the scheduling strategies which succeed in this particular domain and will allow to draw conclusion about the actual involvement/importance of advanced planning techniques in the competition. Finally, a set of proposals will be presented to improve the domain from a planning perspective, in order to make RCLL more appealing for the planning community.

RoboCup Logistics League (RCLL)

The RCLL is part of the RoboCup (Steinbauer and Ferrein 2016) initiative that aims at stimulating research on Robotics and Artificial Intelligence. RCLL is a great testbed for novel approaches for Industry 4.0 that aim to improve flexibility and autonomy. It resembles a production setup with multiple agents that work on individualized orders that arrive on the fly. The idea is that some technology which performs scenarios in RCLL, should be able to perform well in similar real scenarios as well, making the competition a bridge between the research and Industry 4.0 setups. The goal is to achieve a flexible and efficient organization of a workflow of a fleet of robots. In the rest of this section, further details of the competition are described.

A game in the competition comprises two phases and lasts for 4+17 minutes. Two teams play against each other in the same environment. Each team can use a fleet of three mobile robots, which are equipped with a gripper able to grasp and place simple objects. During the first phase, called *exploration phase* and lasting for four minutes, the robots need to explore the environment in order to detect the position and orientation of the so-called stations. The stations are the (static) machines with which the robots interact to progress the assembling of a product. There are several types of them, serving different purposes like fetching raw material, assembling parts, or collecting the final product. Both teams have their own exclusive set of six Modular Production System (MPS) stations. To avoid any advantage for one of the teams and to keep the position of stations unknown until the start of the game, the positions are randomly allocated and symmetric (along the shorter middle axis) with respect to the positions of the corresponding opponent stations. Considering that the robots of the teams start in the two short sides respectively, each team has the same view of the environment, ensuring fairness. An example of the environment can be seen, in a simulation, in Figure 1.

Figure 1: Rendering of the RCLL setting.



During the 17 minutes *production phase*, the robots of a team must respond to received orders by assembling and delivering products. Three orders are immediately requested at the start of the game, and further will be generated during the game. Fairness is preserved by asking for orders of the same complexity to both the teams. Each product consists of some pieces stacked together: a base on the bottom, from one to three rings in the center (depending on the complexity), and a cap on the top. The final product must then be left at a delivery station. There are four types of stations, whose names are quite self-explanatory: base stations, ring stations, cap stations and delivery stations. Since needed pieces have different colors, specified by the order, different machines need to be used as each machine serves only a limited set of colors. Moreover, rings and caps are not given to the robots for free, and the robots must provide supplementary pieces. In Figure 3 the production chain of a medium-complex product is depicted. For each order there is a soft deadline. The reward for an accomplished order depends on both its complexity and its delivery time. Intermediate awards can be given each time a new piece is correctly added to the product, independently by its final delivery. This reward system increases the number of ways and potential strategies a team can adopt to win. Since there is not a requested order of delivering over the products, different approaches can be deployed to achieve as many points as possible. This is an interesting feature of the RCLL from the planning perspective.

The *online* nature of the competition is not only represented by the constant request of new orders, but also by the fact that the robots need to execute their plan in an real environment, to avoid conflicts with the opposing robots, and to react to station's failures. In order to foster such skills, some stations are forced offline for some minutes during the game.

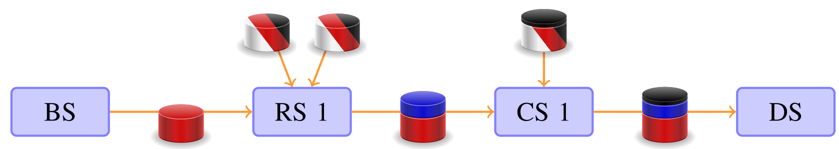
The game is managed by a software called *referee box* (Niemueller et al. 2013) that mimics a Production Management System (PMS). It has different duties, like dispatching orders, collecting and assigning the points achieved by the two teams, communicating with both the robots and the stations, and reporting the actual state of the game.

Figure 2: Robot interacting with a machine.

Winning Architectures and Strategies

In this section we are going to analyze the strategies adopted by the most successful teams of the last years, namely the Carologistics team from RWTH Aachen University, and the

Figure 3: Example production chain for an order of complexity *C1*, adapted from RCLL rulebook (Coelen et al.). This order requires 2 additional workpieces (consumables) at the ringstation (RS1) and a cap loaded at the capstation (CS1).



GRIPS team from Graz University of Technology (Ulz, Ludwiger, and Steinbauer 2019; Hofmann et al. 2019). In particular, we are interested in their approaches used for scheduling and planning, in order to draw some conclusions regarding the design of an optimal planner for this test, as well as to analyze if the actual RCLL setting is really able to stimulate advanced research in planning and plan execution in dynamic environments.

The approach used by GRIPS to scheduling and planning is inspired by hierarchical task network planning (Ghallab, Nau, and Traverso 2004). An order is split into several tasks, which are in turn divided into subtasks that cannot be split further. To keep the representation simple at this level, the task representation is rather abstract, since there is a refinement step performed in the executive running on each robot. The tasks are divided into two classes:

- GET TASK: the assigned robot must navigate to a given station, in order to get a specific workpiece;
- DELIVER TASK: the assigned robot must navigate to a given station to deliver the carried workpiece/product;

At this level navigation concerns are abstracted away and not managed actively. The scheduler simply ignores them, together with the linked issues like conflicts or travelling time, while assigning a task to a robot. A low-level motion planner takes care of the navigation when required, without any consideration in the high-level scheduler.

When a new order arrives from the referee box, in a first step so called critical tasks get identified. Since every workpiece requires a specific color, and there exists only one station which is able to provide workpieces with that color, we can immediately connect each task to the required MPS. While this lack of redundancy is good for performance, as it cuts the search space, it also reduces the number of potential solutions and can be a problem in case of station failures, causing the abortion of the order. We will come back to it in the next sections. The next step is to further split some of the tasks. We know that, for some pieces, the stations require the insertion of supplementary pieces. The subtasks taking care of that are called *resource tasks*. Moreover, some additional tasks can be generated, called *uncritical*, which may speed up the production of future orders. Summing up, subtasks belong to one of the following three categories, in addition to the previous GET/DELIVER classification:

- *Critical Tasks* represent the actual production flow. A failure of a critical task means the failure of the entire product, whose assembling may be restarted;
- *Resource tasks* are responsible of loading a station with the required workpieces for a payment, in order to "unlock" a critical task. In case of failure the task can simply be reassigned;
- *Uncritical tasks* are tasks that do not harm the production at all, but can speed it up if successfully completed.

In the next paragraphs we will describe the scheduling and planning process. We will refer to it just as scheduling process, since it will be clear at the end of this section that there is no proper planning at all.

The assignment process is based on a *request-response* approach. The idea is that the robot asks for a task when it is free. The assignment is then implemented through some greedy hand-coded rules which may apply in the moment a robot is asking for a task. When this happens, three situation can applies:

- **Task in active assembly**: if the assembling of a product is started, the next task of the same order is assigned to the robot, given that its preconditions are fullfilled;
- **Start new assembly**: if the previous case does not apply, the robot just start to process a new order by executing the first task;
- **Dummy task**: in this case, where no feasible task can be found, the robot is assigned to a so-called dummy task such that it is not blocking any relevant MPS while idling.

The Carologistics scheduler makes use of a similar approach. There is no long-term planning and situation classification is performed to select the next action whenever a robot is idling (Niemüller, Lakemeyer, and Ferrein 2013).

More precisely, the goal reasoning approach of Carologistics scheduler split the production of an order in multiple subgoals, comparable to the task concept of GRIPS. Similar to GRIPS subgoals have different priorities, and a robot selects a goal to pursue depending on that. In decreasing order, the priority levels are: **URGENT**, **FULLFILL-ORDERS**, **PREPARE-RESOURCES** and **NO-PROGRESS**. The subplan solving the specific subgoal can be either generated from an external planner or based on a prebuilt database of hand-crafted plans. While the former approach is more flexible, the latter is the one which guarantees the best performance. This approach is used during competitions.

In order to answer to the original question, if the actual setting of the RCLL fosters advanced research in planning and plan execution, the fact that the two most successful participants do not even plan at all, while the scheduling is encoded in a very greedy fashion, raises some doubts about the challenges for the planning community currently posted by the RCLL setting. The fact that rather simple methods are sufficient to master the challenge renders the actual RCLL setup to be less stimulating for the planning community.

Foster Planning in RCLL: Some Proposals

As discussed in the previous sections, there are good reasons to conclude that, in the actual state, the RoboCup Logistics League does not represent a perfect evaluation of setup for planning systems for multi-agent systems in dynamic environments. Although, RCLL is primarily focused on the integration of many fields of robotics and AI to achieve flexible multi-agent system, it is obvious that planning should be an important part of such a system. Right now it does not seem to be the case. In order to backup this claim, would be interesting to find out how much the actual maximum achievable points are and relate them to the points actually achieved by the winner approaches. Since the actual setting provides a rich set of open tasks and idle moments are rare, we suspect that, using the *request-response* approach, will be close to the maximum.

A similar discussion on the role of planning within the RCLL initiative has been made in (Niemueller et al. 2016), with the results presented in (Niemueller et al.). The conclusion of this paper is similar to the one drawn by us, highlighting the fact that this particular competition is more focused on short-term planning and dynamic adaptation, rather than classical planning. A comparison between different planners is shown in the website (CaroASP, CaroSMT and POPF), but without testing them in a game against the greedy schedulers from Carologistics and GRIPS.

In order to create interesting challenges for the planning community and to foster the need for advanced techniques for planning and scheduling we like to propose some changes and extensions to the RCLL setting. We like to stimulate interesting discussion in the community about the possible improvements of the competition in this perspective.

Station redundancy

In the previous sections we have seen that, given a specific workpiece with a specific color, there exists only one station that is able to provide that particular piece. This setting forces some limitations. First, in case of a failure of the station, all the affected orders must be delayed or aborted. Second, having only one station implies there are no choices that need to be done by the planner/scheduler. While this feature improves the solving time of a planner, it also further restrict a domain which is already affected by a lack of choices. Adding additional redundant stations could make the scheduling and planning part more interesting, since we are allowing a richest set of possible solutions. Moreover, it will bring advantages for the online planning and execution part as well. If a station fails, more possibilities are open for the planner. It has to decide if replanning is appropriate or if plan repair is feasible, as well as if it is better to suspend the current orders or to continue it using another station. With such many decisions to make, greedy approaches may be not good enough anymore to reach the results of an advanced planning system.

Scaling up the problem

The simplest way to make the domain more challenging is to just scale up the entire problem, or a part of it. Possible changes are more robots, more stations, a bigger field, more orders to accomplish or an extended production time. Having an higher number of robots and stations allows to work on more products in parallel and as a consequence the scheduler will have more options to maximize its objective function. In this kind of environment, if we also consider a bigger field, the actual *request-response* strategy risks to ignore many possible improvements of the plan.

Consider the case in which an idling robot R_1 is asking for a task. The central server answers with the task T, which for some reasons can be performed, in that moment, only on the station S, very far away from R_1. Imagine that there is a robot R_2 close to S, which will become free one second later. The problem is that when R_2 becomes available the task T has already been assigned to $R1$, causing a makespan's delay equal to the travel time from R_1 position to S. This situation can be avoided with real planning. While this can happen in the actual settings as well, in a scale up problem with more robots this issue becomes more frequent and a bigger field increases the potential delay.

Increasing the production time, from the actual 17 minutes, may render advanced planning approaches more useful than greedy approaches. If right now the improved potential of a plan returned from a real planner is negligible with respect to the *request-response* assignment, these improvements may be amplified by a longer production window.

While scaling up the problem would be the first idea when one wants to make a more complex version of the problem, it comes with some serious disadvantages, in particular in relation of the high computational complexity of applied methods. In fact, complex enumeration techniques like Classical or Temporal Planning, Logic Programming, Constraint Programming and Integer Linear Programming are known to perform poorly for scaled up problems. This makes this proposal less appealing than the others. But a partial scaling up of some features (like the production windows) can still represent a right step towards a better integration of advanced planning methods.

Carrying Multiple Pieces

Another proposal, and maybe the most promising one, is to allow the robot to carry multiple workpieces at the same time. This is a good generalization of the problem, inspired by real-case scenarios. It is in fact quite common, for a smart factory, to have mobile robots with a stack, capable of carrying multiple objects (Fabricius et al. 2020). From the planning's point of view, this feature makes the problem more complex, since a robot now has the capability to work on more tasks at the same time, and being able to save a lot of traveling time. At this point, smart long-term planning of the future is required, since the number of different situations and options becomes too large to be hard-coded and managed by a greedy approach.

The type of media used by a robot to carry more pieces can also influence the outcome. A random-access media would guarantee the maximum level of freedom to the planner and allowing it to minimizing the makespan. However, such kind of container would be pretty difficult to be physically implemented in reality. Other approaches, like FIFO (queues) or LIFO (stacks) containers would be more realistic and may also help the planner to cut the search space earlier, as they introduce more constraints. Stacks in particular are easy to implement and already present in many factories (Fabricius et al. 2020).

Conclusion

This paper aims to foster a discussion about the actual settings of the RoboCup Logistic League and about what changes can be made to make the competition more interesting for advanced approaches for planning, scheduling and plan execution. When the actual settings were first presented in (Niemueller et al. 2015), integration between the AI/planning community and robotics community was one the goal of the RCLL. Yet, it keeps mainly referring to the robotics community. As it can be seen in the discussion of the winning approaches and systems, the most successful tools do not employ planning at all. Lack of choices, together with a overwhelmed quantity of order in a short time window, makes the actual domain less interesting for the planning community, valuing greedy *request-response* strategies which hard-codes the different situations.

As a first step towards the direction of an better integration of planning, we presented some proposal on how the RCLL setting can be made more challenging and attractive for the planning community. Potential improvements are not limited to a simple scaling of the approach but may comprise interesting adjustments of the structure of the problem. We are hoping to get some feedback or new ideas to attract planning researchers into more realistic scenarios with the RCLL.

References

Coelen, V.; Deppe, C.; Hoffmann, T.; Karras, U.; Niemueller, T.; and Rohr, A. Rules and Regulations - RoboCup Logistics League. http://www.robocup-logistics.org/rules.

Fabricius, F.; De Bortoli, M.; Maximilian, S.; Michael, R.; Gerald, S.; and Martin, G. 2020. Towards asp-based scheduling for industrial transport vehicles. In Roth, P.; Steinbauer, G.; Fraundorfer, F.; Brandstötter, M.; and Perko, R., eds., *Proceedings of the Joint Austrian Computer Vision and Robotics Workshop 2020*.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated planning. Theory practice*.

Henning, K. 2013. Recommendations for implementing the strategic initiative industrie 4.0.

Hofmann, T.; Limpert, N.; Mataré, V.; Ferrein, A.; and Lakemeyer, G. 2019. *Winning the RoboCup Logistics League with Fast Navigation, Precise Manipulation, and Robust Goal Reasoning*. 504–516.

Niemueller, T.; Karpas, E.; Vaquero, T.; and Timmons, E. Planning competition for logistics robots in simulation. `http://www.robocup-logistics.org/sim-comp`. Website, Results presentation.

Niemueller, T.; Lakemeyer, G.; Ferrein, A.; Reuter, S.; Ewert, D.; Jeschke, S.; Pensky, D.; and Karras, U. 2013. Proposal for advancements to the llsf in 2014 and beyond. *ICAR–1st Workshop on Developments in RoboCup Leagues*.

Niemueller, T.; Ferrein, A.; Reuter, S.; Jeschke, S.; and Lakemeyer, G. 2015. The robocup logistics league as a holistic multi-robot smart factory benchmark. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2015 (IROS) – Open forum on evaluation of results, replication of experiments and benchmarking in robotics research*.

Niemueller, T.; Karpas, E.; Vaquero, T.; and Timmons, E. 2016. Planning competition for logistics robots in simulation. In *International Conference on Automated Planning and Scheduling (ICAPS) - Workshop on Planning and Robotics (PlanRob)*.

Niemüller, T.; Lakemeyer, G.; and Ferrein, A. 2013. Incremental task-level reasoning in a competitive factory automation scenario. In *AAAI Spring Symposium: Designing Intelligent Robots*.

Steinbauer, G., and Ferrein, A. 2016. 20 Years of RoboCup. *Künstliche Intelligenz* 30(3-4):221–224.

Ulz, T.; Ludwiger, J.; and Steinbauer, G. 2019. A robust and flexible system architecture for facing the robocup logistics league challenge. In Holz, D.; Genter, K.; Saad, M.; and Stryk, O., eds., *RoboCup 2018: Robot World Cup XXII*, Lecture Notes in Computer Science, 488–499. Springer, Cham.