ProSeqqo: Open-source Solver for Robotic Task Sequencing*

László Zahorán[1,2], **András Kovács**[1]

[1] Institute for Computer Science and Control (SZTAKI), Eötvös Loránd Research Network (ELKH), Budapest, Hungary
[2] EPIC InnoLabs Ltd., Budapest, Hungary
{laszlo.zahoran, andras.kovacs}@sztaki.hu

Abstract

ProSeqqo is a generic, open-source solver for process planning and sequencing in industrial robotics. It captures planning problems from diverse applications where task sequencing can be coupled with different types of discrete choices on how the tasks are performed: the selection of the inverse kinematic (IK) solution for a task originally defined in task space; choosing the direction of the motion in cutting or arc welding; or selecting the entry and exit points when following a closed contour. The planning problem can be declared in arbitrary dimensions, e.g., in the 3D task space or in the 6D robot joint configuration space. Moreover, the optimization criterion can be selected from a large set of predefined cost functions, such as minimizing the Euclidean travel distance, minimizing the travel time assuming limited robot joint velocities and accelerations, or a custom cost matrix can be defined. Additional side constraints, such as precedences or resource changeovers can also be captured. The planning problem can be defined using an easily readable problem definition language. This description is transformed into a generalized travelling salesman problem (GTSP) with precedence constraints, and then solved using the algorithms of the Google OR-tools VRP library.

The demo video is available at https://youtu.be/vK53AcCoVT8

Source codes and examples can be downloaded from https://github.com/sztaki-hu/ProSeqqo

Introduction

Task sequencing problems arise in various robotic applications in radically different ways: sequencing must be coupled with different types of decisions on how the tasks are executed; they can be defined in different dimensions, such as the 3D task space or the 6D joint configuration space of the robot; and relevant optimization criteria and side constraints can vary. Accordingly, scientific literature and industrial practice are both dominated by dedicated solution approaches for individual applications. Yet, a thorough literature review showed that the overwhelming majority of these approaches rely on common mathematical methodology, and the arising sequencing problems can be formulated as some extension of the well-known travelling salesman problem (TSP).

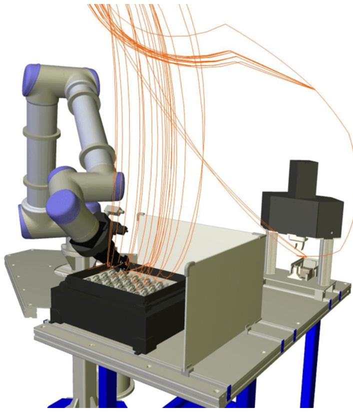

Figure 1: Application to camera-based robotic pick-and-place.

Our objective with the development of ProSeqqo was to offer a generic problem description language, easily editable by domain experts even without a deep background in combinatorial optimization, to capture typical task sequencing problems arising in industrial robotics. The problem descriptions are then transformed into a GTSP, potentially with precedence constraints, and solved using local search techniques of Google OR-tools. This short paper gives a brief overview of the capabilities of ProSeqqo. A detailed presentation of the applied models and algorithms, as well as a thorough experimental evaluation is available in (Zahorán and Kovács 2022).

Modeling and Solving Robotic Task Sequencing Problems

The robotic task sequencing problem can be formulated using a hierarchical representation as follows (see also Fig. 2):

- *Processes* are the largest units of work in the model. The robot must complete a process before moving on to the next process.
- An *alternative* is a possible way of executing a process. Hence, one alternative must be chosen for execution for each process.

---

*This research was supported by the National Lab for Autonomous Systems and the ED_18-2-2018-0006 grant.

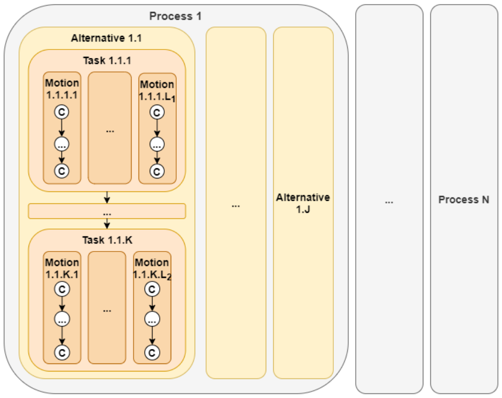

Figure 2: Hierarchy of processes, alternatives, tasks, motions and configurations in the model (Zahorán and Kovács 2022).

- Each alternative consists of a series of *tasks*. Tasks of the selected alternative must be executed in the pre-defined order.
- Each task can be executed using one of the multiple candidate *motions* defined for the task.
- Finally, during each motion, the robot must visit a sequence of *configurations*. Configurations can be defined in arbitrary dimensions.

Accordingly, three types of decisions must be made when solving the planning problem: processes must be sequenced, an alternative selected for execution for each process, and a motion chosen for each task. Alternatives can be applied to describe choices that must be maintained consistently throughout a process, whereas motions capture choices that can be made independently for each individual task. For example, in a pick-and-place application, one process corresponds to each workpiece to be manipulated; alternatives are the possible grasping modes of the workpiece; each alternative contains a picking and a placing task; and motions within a task encode the different IK solutions for the given picking or placing pose.

Precedence constraints can be defined on two levels, between two processes or between two motions. Since a motion may or may not be executed in the resulting plan, the latter is a conditional precedence constraint that expresses a logical relation between the execution mode and the order of the processes. Finally, problems can be *cyclic* or *acyclic*, according to whether the robot must return to its starting configuration, e.g., in a mass production scenario.

ProSeqqo searches for a plan that minimizes the total travel cost along the robot path. The cost can be calculated using one of the various built-in cost functions, such as the Euclidean or the Manhattan distance between subsequent configurations, or the travel time assuming limited joint velocities and accelerations. Resource changeover costs or penalties for idling between effective tasks can also be captured. If needed, custom travel costs can be defined in matrix format. The problem can be encoded in the custom SEQ format (see Fig. 3), in JSON, in XML, or submitted to ProSeqqo via an API, which enables using ProSeqqo in complex planning workflows or in online planning scenarios.

```
Cyclic:                         True
StartDepot:                     1
DistanceFunction:               Euclidean
LocalSearchStrategy:            GreedyDescent

ConfigList:
1; [-1,868; -1,4441; 1,69802; -1,99163; -1,5198;-0,293486];Camera
2; [ 1,572; -0,3077; 0,46288; -4,92137; 2,44224;-1,6411]; Pick1_1
3; [ 1,572; 5,9754; 0,46288; -4,92137; 2,44224;-1,6411]; Pick1_2
4;[ 1,228; 4,6728;-2,04866; -0,40084; 1,72721;-0,2865]; Place1_1
5;[ 1,228; 4,4296;-1,45343; 2,38873;-1,72721; 2,8550]; Place1_2
6; [-1,501; 3,5196;-0,43442; -2,38683; 1,54423;-3,1638]; Pick2_1
7; [ 1,383; -0,3857; 0,45309; -0,78662;-1,79213; 2,9516]; Pick2_2
8;[ 1,228; 4,6728;-2,04866; -0,40084; 1,72721;-0,2865]; Place2_1
9;[ 1,228; 4,4296;-1,45343; 2,38873;-1,72721; 2,8550]; Place2_2

ProcessHierarchy:
0; 0; 0; 1; [1]
1; 0; 0; 2; [2]
1; 0; 0; 3; [3]
1; 0; 1; 4; [4]
1; 0; 1; 5; [5]
2; 0; 0; 6; [6]
2; 0; 0; 7; [7]
2; 0; 1; 8; [8]
2; 0; 1; 9; [9]
```

Figure 3: Sample problem definition in SEQ format from the robotic pick-and-place application.

The above problem definition is transformed into a GTSP with precedence constraints, which can be solved using different local search techniques offered by the Google OR-tools VRP library, such as greedy descend, tabu, or guided local search (GLS). It should be noted that, in contrast to classical precedence-constrained GTSP where precedences are defined between classes, motion precedences correspond to constraints between individual vertices. In such a case, finding an arbitrary feasible solution can be a complex combinatorial problem in itself. Then, ProSeqqo uses a mixed-integer linear programming solution approach to find a feasible plan, which can be submitted to local search as an initial solution for further refinement.

Case Study

(Zahorán and Kovács 2022) present an experimental evaluation in five different applications, including both real industrial applications and scientific case studies. Below, the application to a camera-based robotic pick-and-place cell is presented. A robot must localize and pick one-by-one the parts arriving in bulk to a lighting table, and place them precisely to a part holder for further processing (Fig. 1). The challenge is computing the task sequence and selecting an IK solution for each task near real-time. In experiments with up to 30 parts to pick, ProSeqqo could find close-to-optimal solutions in 0.1–1 s, which enabled the application in online planning scenarios. In particular, GLS provided a consistently good performance. Yet, using the appropriate GTSP representation, from the various formally equivalent representations, was crucial.

References

Zahorán, L.; and Kovács, A. 2022. ProSeqqo: A generic solver for process planning and sequencing in industrial robotics. *Robotics and Computer-Integrated Manufacturing*, 78: 102387.