

The Vehicle Routing Problem with Time Windows applied to a real-world example

Author: Katharina Henggeler

Supervisor: Prof. Dr. Martin Raubal

Advisors: Dominik Bucher, Ioannis Giannopoulos

Bachelor's Thesis, FS 2015

Problem definition and aim

The Vehicle Routing Problem with Time Windows (VRPTW) describes the problem of having a set of customers and a fleet of vehicles that need to deliver goods to each of these customers exactly once. All vehicles must start at a unique depot and after completing their route return to that same depot again. Each customer specifies the number of goods it demands and each vehicle has a limited capacity. In the most basic case all vehicles have the same capacity. Additionally, a time window is defined for each customer. This time window denotes the time interval during which the goods delivery must take place.

The objective is then to minimise the total distance travelled and the number of vehicles needed while still keeping all the constraints mentioned above. Figure 1 also illustrates this.

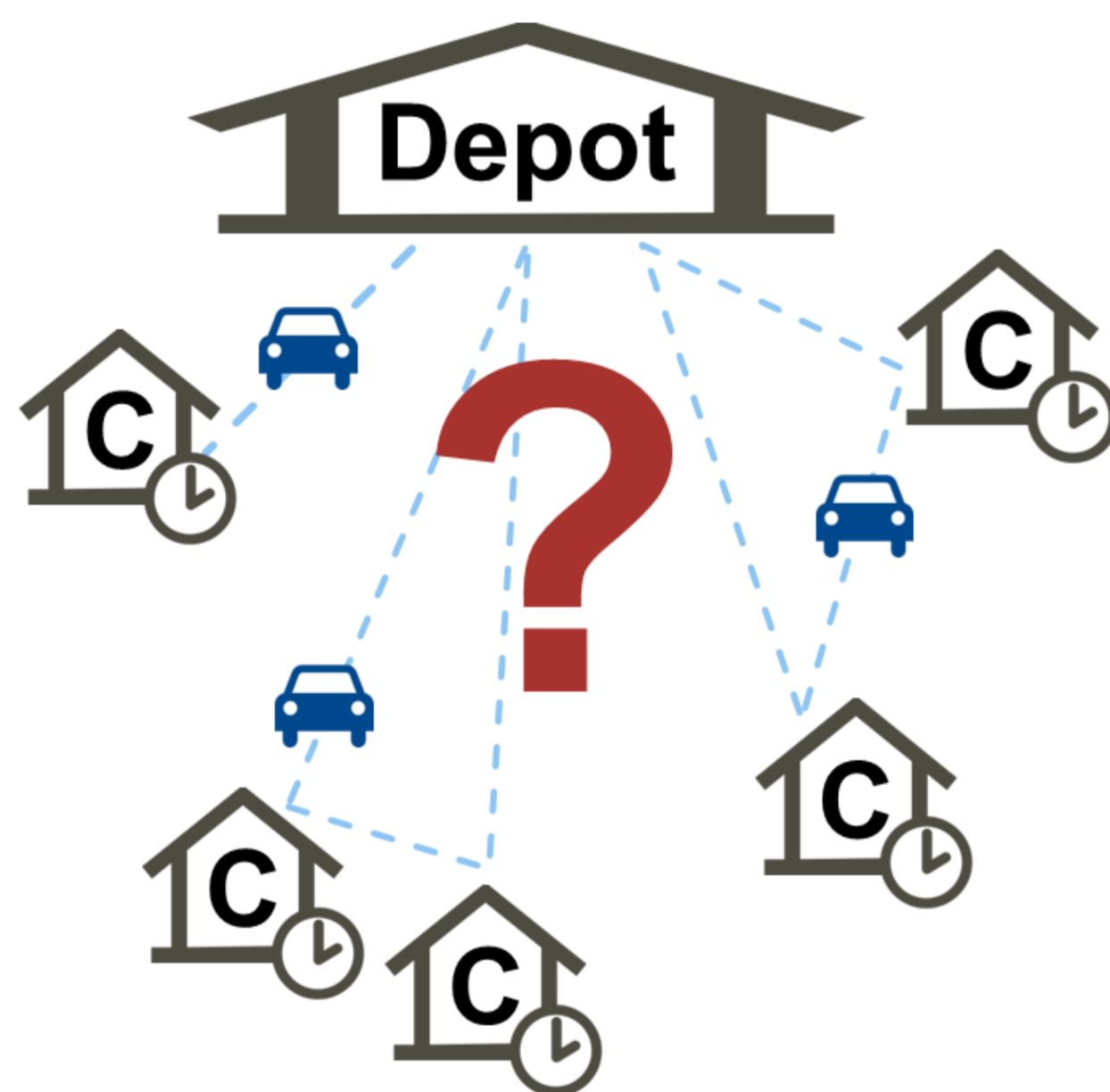


Figure 1: Illustration of the VRPTW

The ZLMSG courier service

The real-world example of the VRPTW that was analysed is the courier service of the Zentrum für Labormedizin St. Gallen (ZLMSG). The ZLMSG provides this courier service for their clients and it retrieves samples that are to be analysed by the ZLMSG. These clients are about 30 hospitals and medical practices in the Nordostschweiz. Determining the optimal routes for the couriers of this courier service essentially means solving a VRPTW, but with certain additional constraints. The most important ones are stated below.

- All couriers must return to the ZLMSG during lunch time (so that the samples can be delivered to the laboratory and analysed right away).
- Some clients need to be visited multiple times a day.
- Clients have preferred visiting times depending on whether they are a hospital or a practice. Some clients wish to be visited during specific time window.

Aim

Thus, the aim was to implement an application to solve this specific real-world example of the VRPTW.

Concept: Application architecture

The developed application essentially consists of three modules:

- The VRPTW Solver module
- The Project module
- The Visuals module

Figure 2 shows the interaction between those modules.

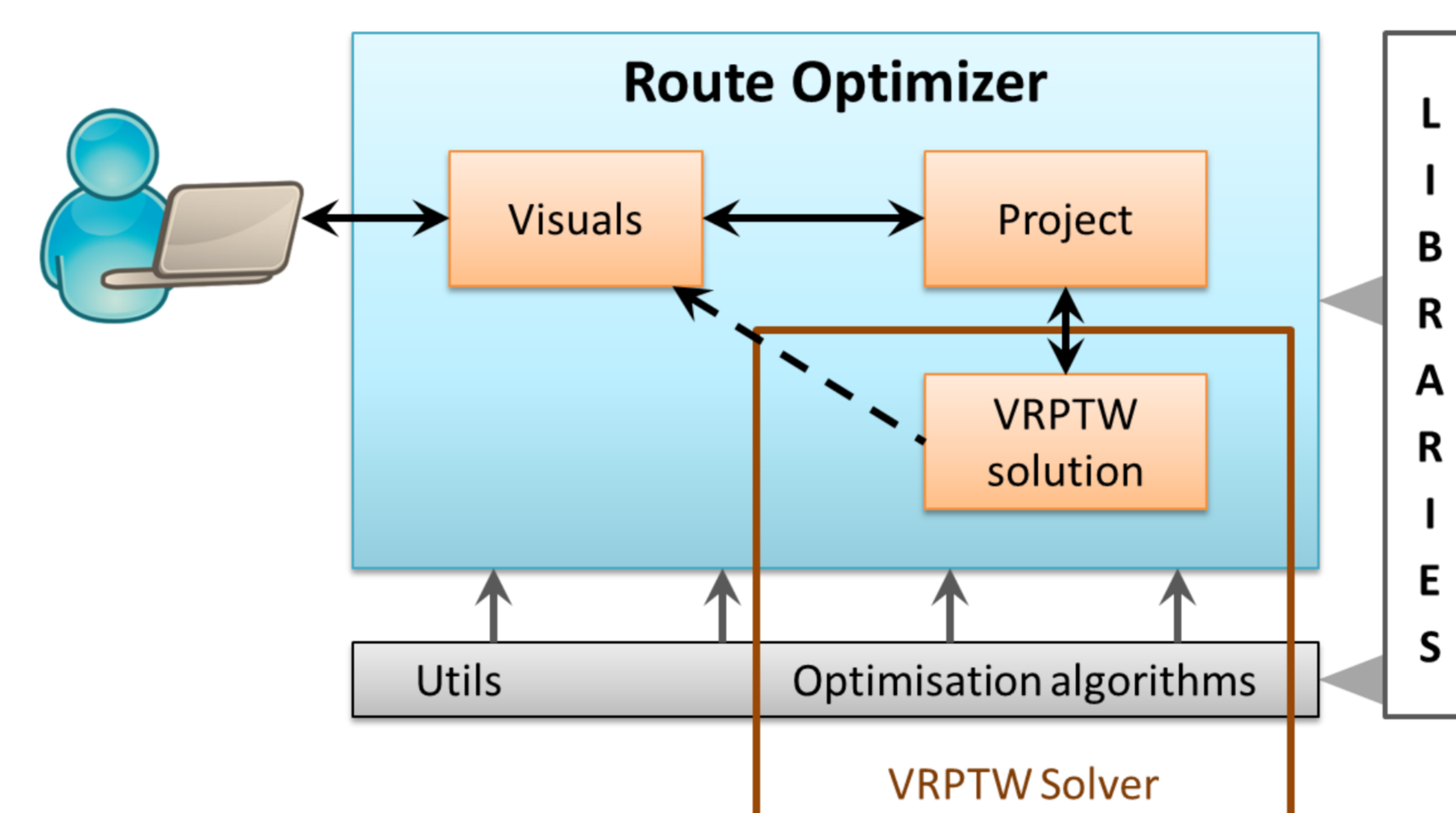


Figure 2: Overview of the application's architecture

The VRPTW Solver module

This module contains everything that is needed to solve a VRPTW. It is again a composition of two separate components. One component provides the functionality, resp. the algorithms, for solving an optimisation problem, and the other component models the solution to be found by the optimisation.

The Project module

The Project module is the actual core of the application. It functions as the interface between the two other modules and is responsible for providing a project file representing the current session and courier service configuration. (See also Figure 3). This courier service configuration, including the additional constraints of the ZLMSG courier service, is modelled with the Problem Model.

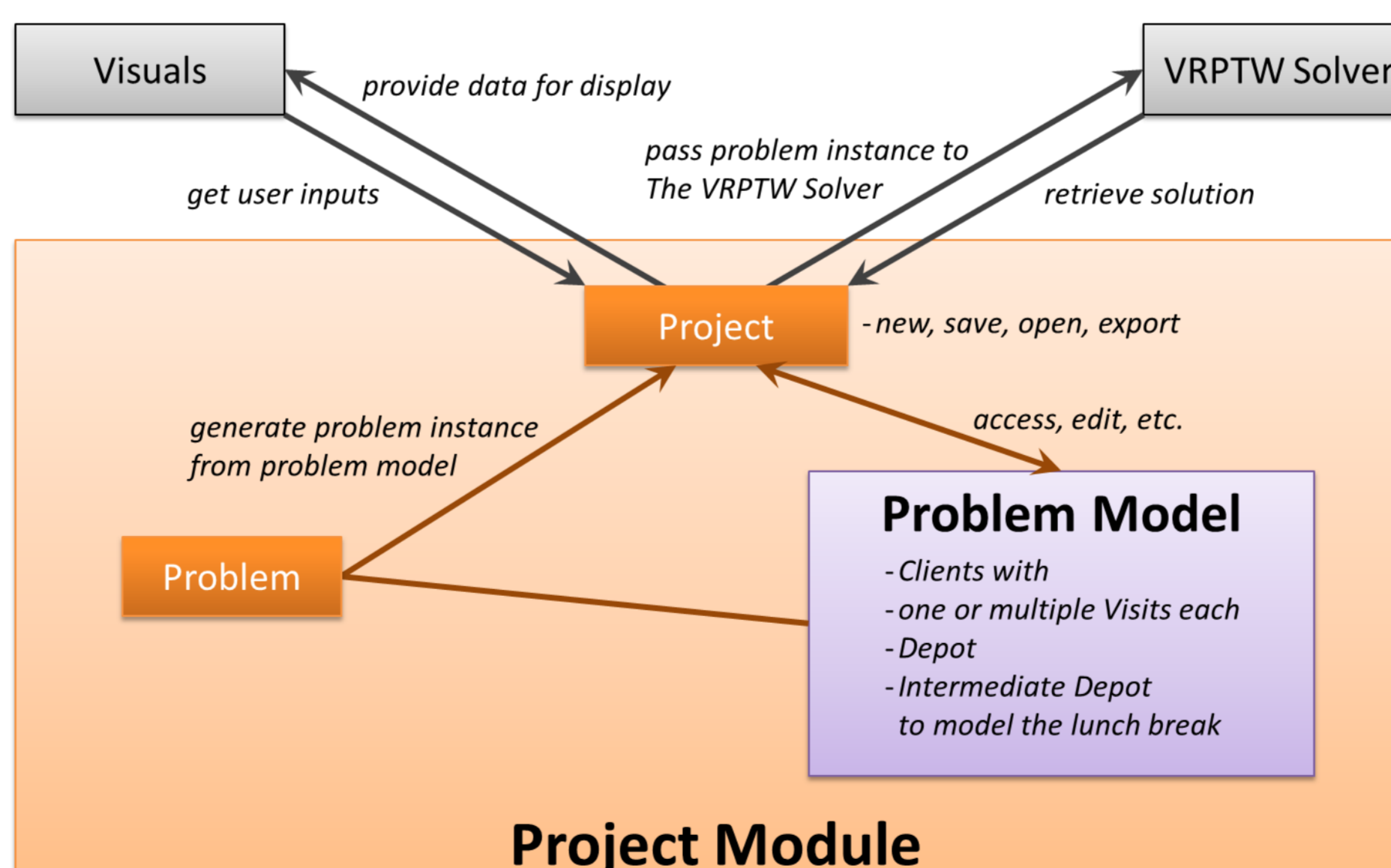


Figure 3: Overview of the Project module and its interactions with other two modules

The Visuals module

The role of the Visuals module is to coordinate the interaction with the user. Thus, the main part of this module is the Graphical User Interface (GUI).

Results: The Route Optimizer

The Route Optimizer is a possible implementation of an application to solve the specific example of the VRPTW that is posed by the ZLMSG courier service. Figure 4 shows a screenshot of the Route Optimizer's GUI.

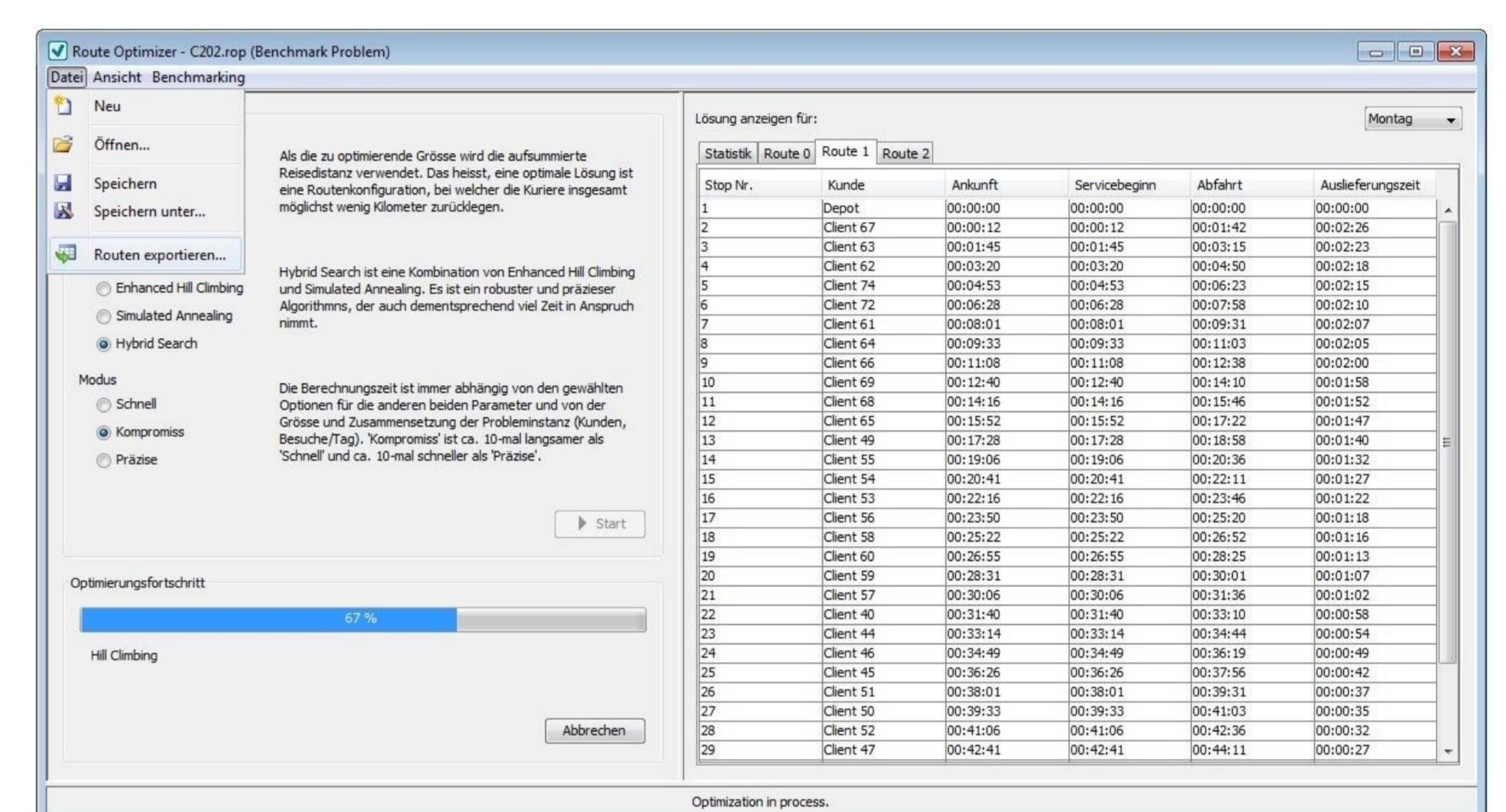


Figure 4: Screenshot of the Route Optimizer application during a running optimisation.

The project files provided by the Project module can be saved and opened again. The Route Optimizer also supports creating new empty projects. In addition to optimising real-world problem instances, benchmark problem instances can be loaded into a new project and processed with the same optimisation algorithms. In order to give the user a choice and to give the application certain flexibility, the user may choose between three optimisation methods, three measures to be optimised, and three modes. When running the optimisation, the Route Optimizer then displays the progress in a progress bar and also offers a possibility to cancel the optimisation. Finally, the found solution, resp. its route schedules, is displayed in a set of tables, one for each route plus an additional table for displaying general information about the solution. These tables can also be exported for further processing and visualisation.

Outlook

Although the Route Optimizer is able to complete the tasks it was designed for, it still has plenty of room for improvement. Especially in terms of usability and application design, there were many issues that could not be considered. The same goes for issues concerning the modelling of the constraints of the ZLMSG courier service.

Also, the Route Optimizer has not yet been tested with respect to usability or usefulness. Hence, this would definitely be one of the first issues to look at.

Additionally, an app could be implemented. It would access the route calculated for a certain courier and display it on a map. This would also offer the possibility to send a short message to the ZLMSG as soon as a courier falls behind schedule.