

# Vehicle Routing with Cross-Docking

Hanne L. Petersen, Stefan Røpke  
{hlp,sr}@transport.dtu.dk  
DTU Transport, Bygningstorvet 115, 2800 Kgs. Lyngby

As part of the current trend towards optimisation of freight transport operations, cross-docking has played an increasing role in recent years, as a means of obtaining better utilisation of vehicles, thereby allowing for a reduction of total driven distance. In order to make efficient use of a cross-dock it is necessary to have access to good data for the transported goods, and proper procedures for handling these data and planning the daily operation. In this paper we present an application of a transportation problem with cross-docking, and our suggested solution approach for solving the corresponding vehicle routing problem.

## 1 Background

The problem to be considered in the following is a pickup and delivery problem, given by a number of orders/requests, where each request specifies an amount of goods to be transported from a given pickup location to a given destination. Each customer site to be visited has a certain time window where it is open. The aim is to construct a set of vehicle routes, such that all requests are served using the available fleet of vehicles, and all visits are carried out within their specified time window. The cost of a given solution depends on driven distance, total time spent, required handling operations, etc., and the purpose is to find the cheapest possible set of routes, to ensure that the transport operator can serve all customers at the lowest possible cost.

In the present application, called the vehicle routing with cross-docking, one or more cross-docks are available to help carry out the required operations. A cross-dock allows for transfer of goods between vehicles, and means that one request can be served by two vehicles in combination, with one vehicle performing the pickup and the transportation of the goods to the cross-dock, and the other vehicle transporting the goods from the cross-dock to the final destination. With multiple cross-docks, yet another vehicle may carry out transport of goods between cross-docks. Thus the cross-dock allows for consolidation of the goods, and thereby more efficient use of the vehicles. Goods from several different origins can be combined to fill up a truck, if their destinations are located close together.

Figure 1 shows an example of how vehicle routes could be constructed using a cross-dock, and Figure 3 shows an example of a cross-dock layout. In Figure 1 the routes are covered by 3 vehicles, and the left side of the figure shows the pickup and delivery routes for each vehicle (“suppliers” are customers who require a visit to pickup goods, and the “customers” are the consumers, where the goods must be delivered). The transfer of goods between vehicles on the right. As can be seen in Figure 3, a cross-

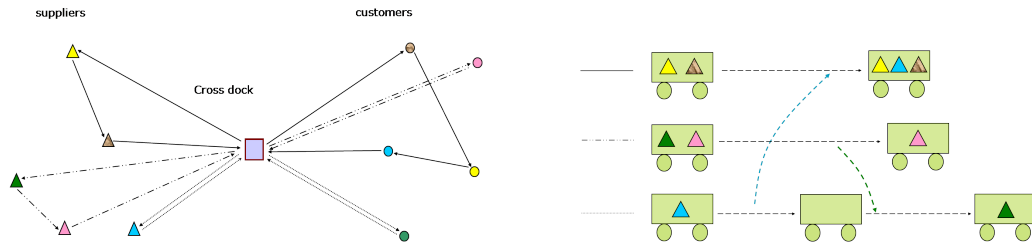


Figure 1: A sample routing plan using a cross-dock.

dock contains one or more areas that are laid out for short-term storage of goods,

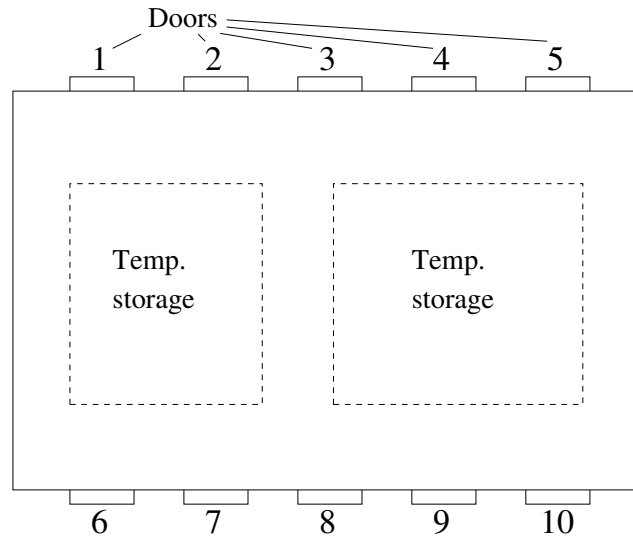


Figure 2: Example of cross-dock layout.

from its being off-loaded an incoming vehicle, and until the re-load and departure of the other vehicle, but in general it should not be seen as a storage opportunity, and as a rule of thumb the cross-dock should be empty at the end of the working day. This underlines the function of the cross-dock as a tool to be used in transportation planning, but not a long-term storage facility. This is also in line with recent focus on the just-in-time principle in logistics, where storage/inventory time should be reduced. Finally, as we shall see, the real-life application leading to this study, is in the business of transporting fresh plants and flowers, which are of course of limited shelf life, and have specific requirements for temperature and watering if they must be stored.

The vehicle routing problem with cross-docking is a special case of the much-studied general Vehicle Routing Problem [2]. The vehicle routing problem is a problem of planning a set of routes to visit a set of customers. It usually deals with distribution of one commodity to all customers, and hence doesn't have the complication of connected pickup and delivery customers. The vehicle routing problem may or may not have time windows determining when each customer can be visited, and may or may not have capacity limitations on the vehicles (some types of goods take so little loading space that capacity is not an issue in distribution planning).

The use of cross-docks in distribution, leads to several interesting planning problems, not just from a vehicle routing perspective, but also regarding the layout and operations of the cross-dock facility itself. This includes assignment of trucks to doors,

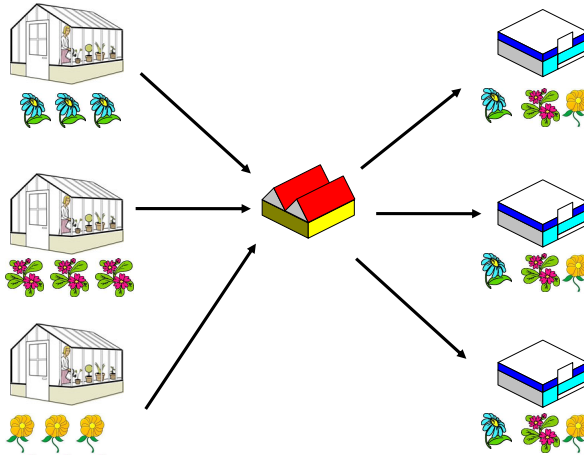


Figure 3: Example using a cross-dock to supply different products.

assignment of the storage areas, handling of goods inside the cross-dock, location of cross-docks, and many other problems, some examples can be found in [1]. However, in this project we will focus exclusively on the associated routing problems.

### 1.1 The company

The application behind the project originates with Alex Andersen Ølund A/S (AAØ), a major Danish transporter of flowers, with activities reaching most of Northern and Mid-Europe, and spreading into other areas of goods transportation as well. The national distribution of flowers, which is being considered in this project, is centered around 3 cross-docks spread throughout Denmark, and covers some 1,000 requests to be served each day by a fleet of 170 trucks.

The distribution includes pickup of flowers at gardeners' greenhouses, and delivery at florists and supermarkets. The greenhouses are typically located in the Western parts of Denmark, while the customers are all over the country, with a higher concentration in the East. This opens good opportunities for use of cross-docking, by bringing the flowers from the greenhouses in to the cross-dock, where repacking takes place, and the flowers can then be sent on to their final destination. The customer locations are distributed all over Denmark, as can be seen from Figure 4.

The temporal distribution of the time windows in which the customer locations can be visited in the preliminary data set is illustrated in Figures 5. All time windows for pickup close at 20.00, so the width of the time windows can also be seen from the figure. It can be seen that the vast majority of pickup operations must be carried out between 17.00 and 20.00, while 17% can be commenced before 16.00.

The delivery time windows do not have a common closing time, however there is limited variation in the opening time. 85% of delivery time windows open at 7.00, 7.75% at 6.00 and 5.75% at 8.00, leaving less than 1.5% of the time windows to open at other times, mostly in the early hours of the morning. The width of these time windows can be found in Figure 6. Most of the time windows are fairly wide at five hours or more, but almost 5% of deliveries have a time window of less than two hours.

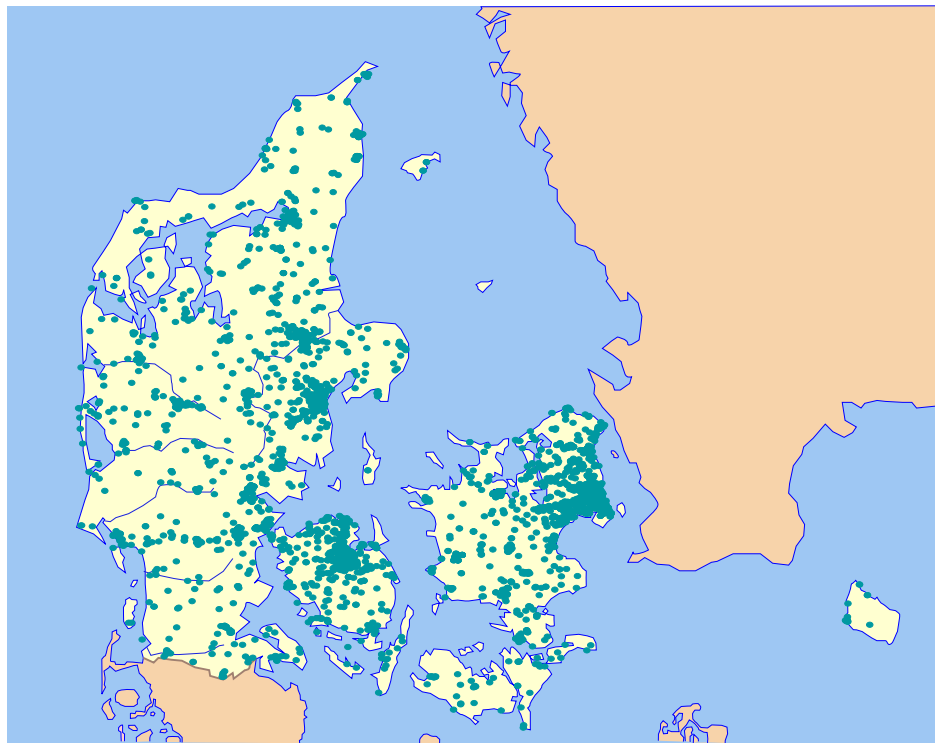


Figure 4: Distribution of customer locations in Denmark.

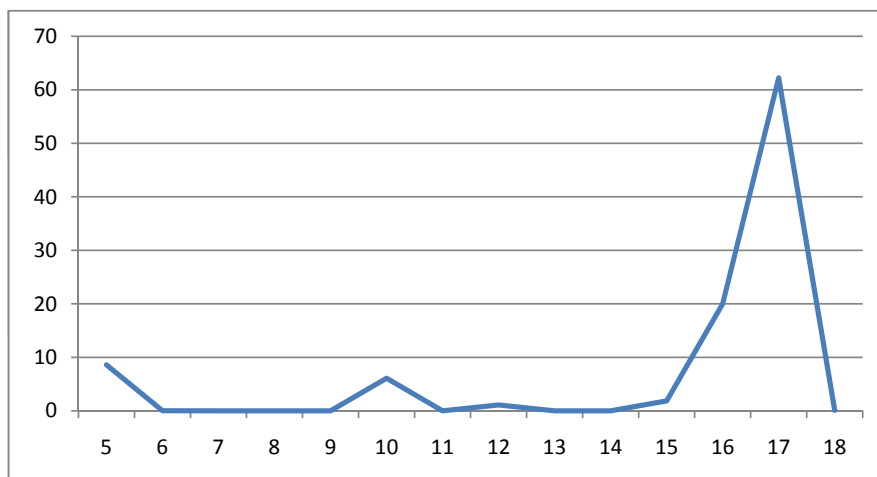


Figure 5: Percentage of pickup operations with a time window starting at a given time of day.

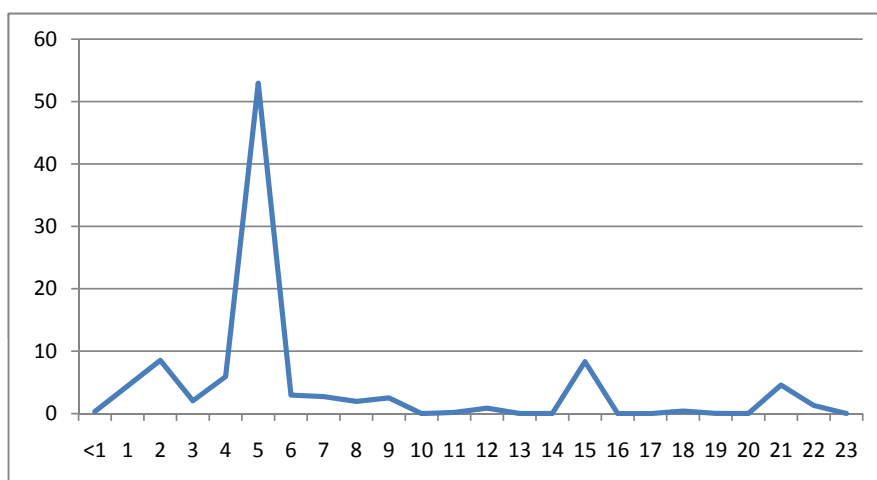


Figure 6: Percentage of delivery operations with a given width of the time window.

## 2 Project Goal

The goal is to produce a prototype for a piece of software that can assist the planner in charge of making the routes. Currently, the routes are planned by humans, and we wish to explore the opportunities for using computer power to assist the human planner. In recognition of the complex structure of the problem and the intricacies of the data (there might be additional restrictions that can not easily be expressed, or other objectives to consider, that cannot easily be converted when looking for the “cheapest” route), we do not expect the software to handle all the “soft” requirements of the problem. Instead we want to produce a tool that can suggest good routes to the planner, but let the planner accept or reject these routes based on other criteria.

For each day of operation, we must be able to present a set of routes, that ensure that all required requests are served on that day. The solution must specify which and which route each truck must drive, such that all customers are visited, and which cross-docking operations will be required, such that all requests can be served in a timely fashion.

The approach consists of two parts: a set of algorithms to calculate a set of near-optimal routes, and an interface that presents the routes to the planner, and allows for the planner to apply certain changes to the solutions. This interface should also allow the planner to make modifications to the data, such as requests and equipment properties.

The solution approach to the routing problem is based on a combination of some well-known methods from operation research, using construction and improvement phases to determine good solutions, and the application of a Large Neighbourhood Search approach [3] in the improvement phase.

During the construction of the graphical user interface it is necessary to consider which informations should be displayed to the user, and how the interface can support the user when making the final routing decisions. The displayed data consist of a combination of tabular data, and maps with a graphical representation of the suggested routes.

## 3 Solution approach and outcome

The solution consists of two parts: An algorithm to generate a “good” set of cheap routes, and an interface to present these solutions to the planner.

### 3.1 The route generation algorithm

The solution algorithm used for generating the vehicle routes is based on the *Large Neighbourhood Search* introduced by Shaw in 1998, and successfully applied to many classes of problems since.

The idea of the Large Neighbourhood Search (and of many similar approaches) is to start from an initial set of routes, and then step-wise perform modifications to these routes, and thereby gradually obtain better solutions. Due to the power of modern computers, such an approach can lead to very good solutions, because a

huge number of potential new solutions can be examined in very short time.

The characteristic feature of large neighbourhood search is the way in which these new potential solutions are generated. This is done by “destroying” the initial solution, by removing a number of requests from the solution, and thereafter “repairing” it by reinserting these requests one by one. At the reinsertion step some requests might be inserted in the same route they were removed from, but many customers might instead be served by different vehicles, on routes where they fit better.

This way we can obtain a lot of solutions that slightly differ from the original, and we can then look at the cost of each of these solutions, to choose one that will be the basis for the next destroy-and-repair step. By repeating this procedure thousands of times it is possible to produce very good solutions.

### 3.2 The user interface

The user interface contains overview screens for both the input and output data. Input data that can be viewed include list of customers and requests to be served, but also some parameters that allow the user to affect the operation of the planning algorithm. The viewing of the output data is more complex and must illustrate the calculated routes in an intuitive way, in the form of both maps and lists, that allow the user to evaluate the solution and look for improvements that lie outside the search scope of the algorithm.

Figures 7 and 8 are screenshots from the developed software using some smaller data sets, and give a general impression of the way the solution.

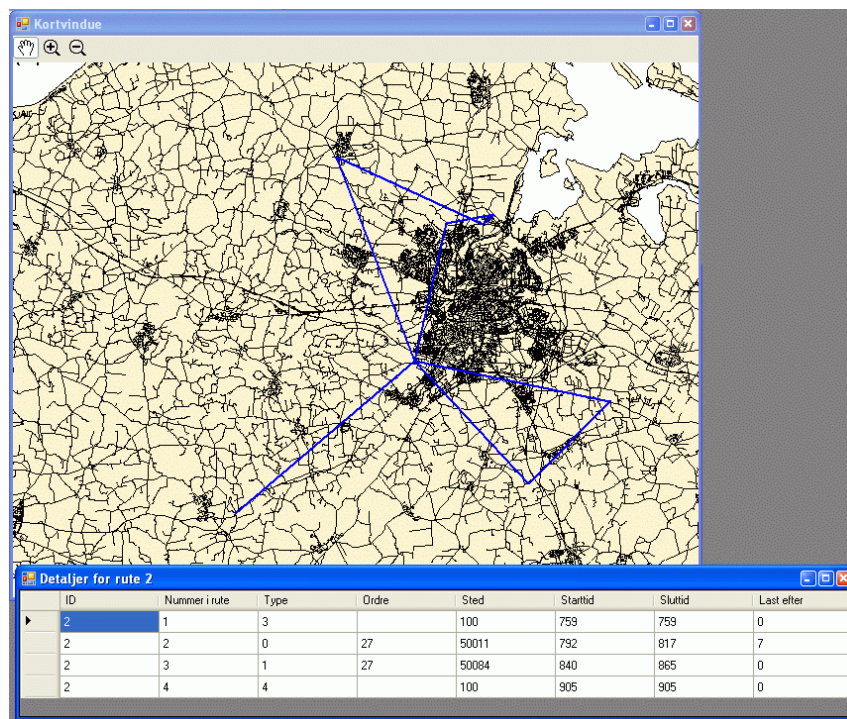


Figure 7: Screenshot showing results from a small data set. The interface shows the calculated routes on a map, and details for each route are presented in separate windows.

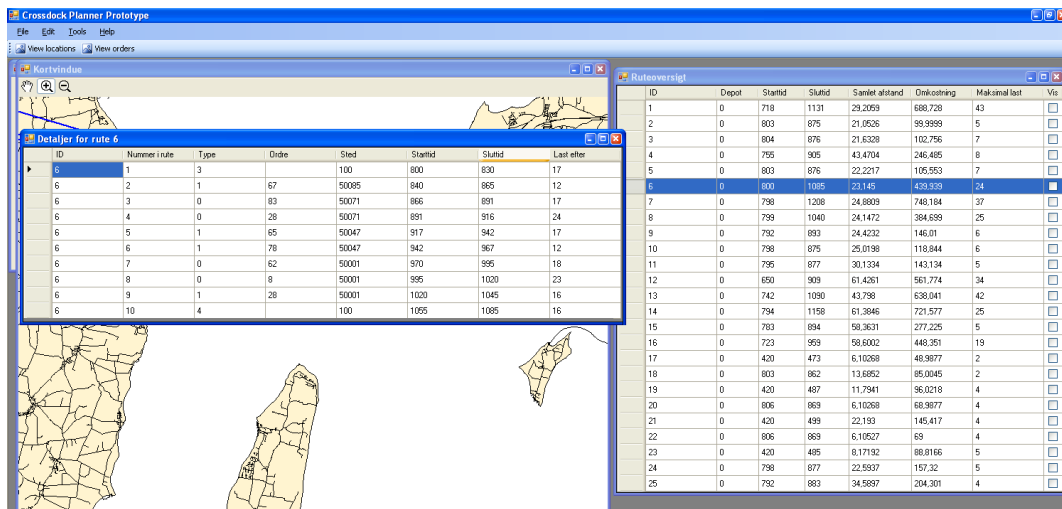


Figure 8: Screenshot showing a list of the calculated routes and a more detailed list of individual stops of one selected route.

The presentation of the solution consists of a map that can be zoomed and scrolled, which can show all routes or a selection hereof. This overview is accompanied by a list containing basic data for each route, such as cost, total distance, maximum loading degree, starting time, duration, etc. This list also allows the user to toggle the viewing of each route on or off. A further list view gives access to more detailed information about each route, detailing each stop, visit time, partial driving times, order references, amount of goods to be loaded at each stop, etc., and again gives access to the order detail view used to present input data, in order to handle customer visits on the order level.

## 4 Conclusion

The usage of routing software can often lead to considerable savings, obtained by reduced driven distance and better use of resources (vehicles and cross-docks). Compared to the traditional planning approach, based exclusively on human planners, a computer-supported approach has the advantage of being able to check many potential solutions in a much shorter time. As the problem size/number of requests increases it becomes still more difficult for the human planner to maintain the overview of all aspects of the planning process, while the computer is much better suited for handling large amounts of data. On the other hand, planners often have a profound knowledge of the system, and this knowledge is very difficult to transfer to the planning software. Therefore, we find that the best overall solution is obtained by using the routing software as decision support for the human planner – preferably by allowing the planner to interact with the software during the final adjustments of the suggested routes.

In this paper we have presented the vehicle routing problem with cross-docking. We have described a real life application where the problem is encountered, and finally we have presented a prototype of a planning tool to solve the problem, and the route generation techniques it applies.



## References

- [1] N. Boysen, M. Fliedner: *Cross Dock Scheduling: Classification, Literature Review and Research Agenda*, Technical Report, 2009.
- [2] Golden, B.L., Raghavan, S., Wasil, E.A.: *The vehicle routing problem: latest advances and new challenges*, Springer, 2008.
- [3] Shaw, P. Maher: *Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems*, Principle and Practice of Constraint Programming—CP98, Springer-Verlag, 1998.