Part I

Approximation Algorithms for Conflict-free Vehicle Routing on Bidirectional Networks

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Chapter 1

Introduction

In the first part of this thesis, we investigate the conflict-free routing of vehicles through a network of bidirectional guideways. Conflicts are defined in a natural way, i.e., vehicles cannot occupy the same resource at the same time, hence forbidding crossing and overtaking. The task is to find a routing consisting of a route selection and a schedule for each vehicle, in which they arrive at their destinations as quickly as possible.

Such conflict-free routing algorithms are needed in various applications in logistics and transportation. A prominent example is the routing of Automated Guided Vehicles (AGVs). AGVs are often employed to transport goods in warehouses (for survey papers we recommend [GHS98, Vis06]), or to move containers in large-scale industrial harbors [SV08]. The guideways can be tracks or any sort of fixed connected and bidirectional lane system. Other related application settings are the routing of ships in canal systems [PT88], locomotives in shunting yards [FLKH05], or airplanes during ground movement at airports [GBM$^+$02, ABR10].

Conflict-free vehicle routing problems can be divided into online problems, where new vehicles with origin-destination pairs are revealed over time, and offline problems, where the vehicles to route are known in advance together with their origin-destination pairs. Here, we concentrate on the offline problem, which is also often a useful building block for designing online algorithms.

Algorithms for conflict-free routings either follow a *sequential* or *concurrent* routing paradigm. Sequential routing policies consider the vehicles in a given order, and select a route and schedule for each vehicle such that no conflict occurs with previously routed vehicles (see [KT91, MKGS05, KJR07] for sequential routing examples in the context of AGVs). Concurrent approaches take into account multiple or all vehicles at the same time. Whereas the higher flexibility of those approaches opens up possibilities to obtain stronger routings than the sequential paradigm, they usually lead to very hard optimization problems. Furthermore, they are often difficult to implement in practice.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Typically, the routing problem is modeled as an Integer Program (IP) which is tackled by IP solvers [Oel08], column generation methods [FLKH05] or heuristics without optimality guarantee [PT88, GBM$^+$02, KBK93].

Sequential algorithms are thus often more useful in practice due to their computational efficiency but suffer from the difficulty of finding a good sequence to route the vehicles. Furthermore, the theoretical guarantees of these approaches are often weak. The goal of this work is to address these shortcoming of sequential routing algorithms. Most of the results presented in the following are also published in [SZ11].

1.1 Problem Formulation

We consider the following problem setting which captures common structures of many conflict-free vehicle routing problems.

Conflict-Free Vehicle Routing Problem (CFVRP). *Given is a undirected connected graph $G = (V, E)$, and a set of k vehicles Π with origin-destination pairs (s_π, t_π) for all $\pi \in \Pi$. Origins and destinations are also called* terminals*. A discretized time setting is considered with vehicles residing on vertices. At each timestep, every vehicle can either stay (wait) on its current position or move to a neighboring vertex. Vehicles are forbidden to traverse the same edge at the same timestep, also when driving in opposite directions, and no two vehicles are allowed to be on the same node at the same time. A routing not violating the above rules is called* conflict-free*. The goal is to find a conflict-free routing minimizing the* makespan*, i.e. the number of timesteps needed until all vehicles reach their destination.*

The CFVRP is a natural first candidate for modelling and analyzing routing problems in a variety of contexts. Clearly, it omits application-specific details and makes further simplifying assumptions.

As a relaxation of the conflict definition above, we assume that vehicles can only be conflicting while in transit, i.e., no conflict is possible before departure and after arrival. The *departure time* of a vehicle is the last timestep that the vehicle is still at its origin, and the *arrival time* is the earliest time when the vehicle is at its destination. We call this relaxation the *parking assumption.* The parking assumption is natural in many of the listed applications since the terminal node occupations are often managed by separate procedures. In AGV systems the dispatching (task assignment) is usually separated from the routing process and takes care of terminal node occupations. In airport

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

ground movement problems, airplanes are assigned to runways and gates before airplane routing starts. When routing ships through a canal system, the terminals represent harbors with usually enough space for conflict-free parking of all arriving and departing vessels.

1.2 Related Work

The model setting investigated in [KBK93, Spe06, Ste08] is very similar to the one used here. The differences lie mostly in the modelling of waiting vehicles, which block edges in their setting. In [KBK93], only designated edges can be used for waiting. However, these variations do not significantly change the problem, and the results can easily be transferred. The main reason why we use the CFVRP setting introduced above is that it leads to a simplified presentation of the algorithms.

CFVRP has many similarities with packet routing [Sch98, PSW09], where the goal is a conflict-free transmission of data packets through cable networks. The crucial difference is that the conflict notion in packet routing is relaxed. It allows for several packets to occupy a node at the same time, as nodes represent network routers with large storage capacity. The concept of edge-conflicts is essentially the same as in the present setting and models the limited bandwidth of the transmission links. Hence, the CFVRP setting can as well be seen as a packet routing problem with unit capacities on every node.[1]

Some sequential routing approaches. We briefly discuss some variants of sequential routing schemes, emphasizing on approaches used later when presenting the algorithms.

The presumably simplest approach is to serially send one vehicle after another on a shortest route to the destination, such that a vehicle departs as soon as the previous one has arrived. The obtained makespan is bounded by $k \cdot L$, where L is used as the maximum origin-destination distance over all vehicles. Since L is a lower bound on the optimal makespan OPT, this is a k-approximation. Interestingly, for general graph topologies, no efficient algorithm was known to substantially beat this approach, i.e. with a $o(k)$ approximation guarantee.

[1]There are approximation results for packet routing with buffer size 1 in [adHS95]. However, contrary to the CFVRP, they consider bidirectional edges on which two packages can be sent concurrently in opposite directions.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Still, several stronger routing paradigms are known and commonly used in practice. An improved sequential routing policy, which we call simply *sequential routing*, is the following procedure as introduced in [KT91, MKGS05]. Vehicles are considered in a given order, and for each vehicle a route and a schedule (timetable) is determined with earliest arrival time, avoiding conflicts with previously scheduled vehicles. For a fixed ordering of the vehicles, a sequential routing can be obtained efficiently, e.g. by finding shortest paths in a time-expanded graph. Sequential routing is often applied with given origin-destination paths for all vehicles, in which case the task is only to find a schedule for each vehicle that determines how to traverse its origin-destination path over time.

For given origin-destination paths, the following restricted version of sequential routing algorithms, called *direct routing*, often shows to be useful. In direct routing, see e.g. [BMIMS04] for the corresponding approach in packet routing, vehicles are not allowed to wait while in transit, i.e., once a vehicle leaves its origin, it has to move to its destination on the given origin-destination path without waiting. An advantage of direct routing is that vehicles only block a very limited number of vertex/time slot combinations.

Combining this concept with the sequential routing, the *direct sequential* algorithm is obtained. Here an ordering of the vehicles is given, as well as a source-destination path for each vehicle. Considering vehicles in the given ordering, the routing of a vehicle is determined by finding the earliest possible departure that allows for advancing non-stop to its destination on the given path, without creating conflicts with previously routed vehicles.

When fixing the origin-destination paths to be shortest paths, sequential routing and its direct variant perform at least as good as the trivial serial algorithm. However, for unfortunate choices of the routing sequence, one can observe that the resulting makespan of both approaches can still be a factor of $\Theta(k)$ larger than the optimum (see [Ste08] for details).

Further related results. Spenke [Spe06] showed that the CFVRP is NP-hard on grid graphs. The proof implies that finding the optimal priorities for sequential routing is also NP-hard.

Polynomial routing policies with approximation quality sublinear in k are known for grid graphs. Spenke introduces a method for choosing a routing sequence with a makespan bounded by $4\mathrm{OPT} + k$. An online version of the problem was investigated by Stenzel [Ste08], again for grid topologies.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Computational results published in [Ste08] indicate that sequential algorithms can have bad performance when the number of route choices of near-shortest-path lengths are limited. For grid topologies, the above-mentioned algorithm of Stenzel [Ste08] takes advantage of the fact that grid graphs contain at least two disjoint routes of almost the same length for each pair of vertices.

1.3 Outline

On the negative side we present in Chapter 2 hardness results showing that there is not much hope to obtain exact solutions even for seemingly simple settings. The results give a theoretical explanation for the difficulties encountered in practice when looking for good orderings for sequential routings.

On the positive side, we consider in Section 3.1 the CFVRP problem on trees, and present a priority ordering of the vehicles leading to a direct routing algorithm with a makespan bounded by $4\text{OPT} + k$. This is achieved by dividing the vehicles into two groups, and showing that each group admits an ordering which leads only to very small delays stemming from vehicles driving in opposite directions.

For general instances, without restrictions on the graph topologies, we show how the tree algorithm can be leveraged to obtain a $O(\sqrt{k})$-approximation, thus leading to the first sublinear approximation guarantee for the CFVRP problem. A crucial step of the algorithm is to *discharge* high-congestion vertices by routing vehicles on a well-chosen set of trees. The purely multiplicative approximation guarantee is obtained despite the $+k$ term in the approximation guarantee for the tree algorithm by exploiting results from the packet routing literature. More precisely, using an approach of Srinivasan and Teo [ST97], we determine routes for the vehicles with a congestion C bounded by $C = O(\text{OPT})$, and never route more than C vehicles over a given tree. These results are presented in Section 3.2.

Additionally, an efficient randomized method with makespan $O(\log^3 k)\text{OPT}+k$ is presented for general graph topologies in Section 3.3. This approach relies on obtaining strong tree embeddings in a compacted version of the graph, therefore avoiding a dependency of the approximation guarantee on the size of the graph, which would result by a straightforward application of tree embeddings.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.