An Application for Simulations at Large Pickup and Delivery Service Providers

Curt Nowak, Klaus Ambrosi, Felix Hahne
{cnowak, ambrosi, hahne}{@bwl.uni-hildesheim.de}

Abstract: Over time, large scale pickup and delivery providers acquire considerable amounts of data on customer orders, including order times, amount and configurations of freight, as well as the locations for pickup and delivery. If their fleet is equipped with tracking devices, additional GPS data can be provided and actual travel routes can be saved for later evaluation. In this paper, we present two use cases for a discrete event-based simulation in the context of transportation companies. We also discuss ways to use company-internal data in order to simulate near-realistic scenarios that build the foundation for internal education and training of human dispatchers and that can at the same time be used for benchmark tests for different allocation and scheduling strategies.

1 Introduction and Motivation

The field of vehicle routing problems (VRP) has been widely researched. Algorithms for optimal solutions as well as essential (meta) heuristics (see e.g. [Dom07, Ohr08]) have been established and there are constructed de facto standard benchmarks that new approaches can be tested against (e.g. [SBS88]). A related problem is the pickup and delivery problem (PDP), in which each transport request is made up of two locations; one for pickup and one for delivery. It contains the VRP as a special case if all pickup locations start and end at a depot. [SS95] have formulated the General PDP (GPDP) in order to create a model that comprises most practical pickup and delivery situations including split loads, reloads, multiple vehicle types, and time windows. Solutions to the GPDP always consist of both allocation of vehicles to freights or vice versa and of the scheduling of vehicles. Allocation and scheduling influence each other and depending on time windows, allocations may imply scheduling as well.

Conventional search strategies for optimal solutions both of the VRP and the (G)PDP are often based upon deterministic problem instances, i.e. all transportation requests and all distances between pickup and delivery locations are known ex-ante. Real-life pickup and delivery providers, in contrast, face a continuous flow of incoming customer orders. Often, response time for allocation and scheduling of urgent orders must be extremely low. Consider the automotive sector, for example, in which late delivery of critical parts can easily cause very expensive production downtimes. Orders can also change over time, e.g. they can be updated or even canceled, and unpredictable external events like traffic jams additionally interfere with routing plans.

Due to these real-life restrictions, transportation companies have no choice but to strongly rely on heuristic approaches for allocating and scheduling their vehicles. Many details still need to be dealt with by human dispatchers and their professional experience. This applies even more to companies that operate on a per-call basis and do not run large transshipment centers. Decision support systems (DSS) help human dispatchers to visualize the current order situation.[1] They can provide a recommendation engine (that in turn relies on above mentioned heuristics) that is able to quickly suggest favorable dispatches, i.e. it recommends allocations for vehicles to freights and vice versa and may additionally support vehicle scheduling. Figure 1 shows a prototypical setup for a transportation company.

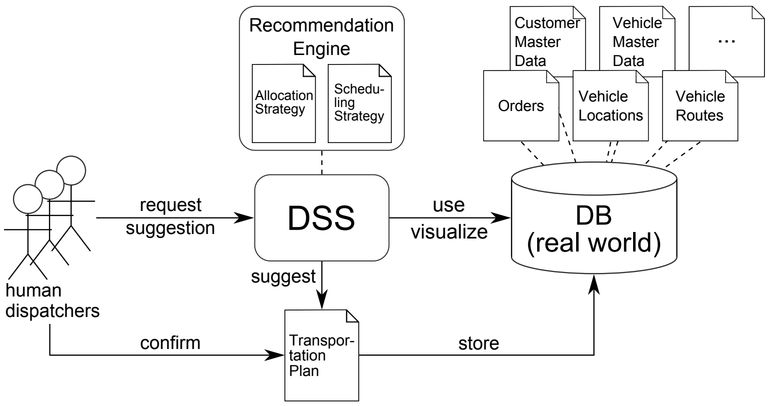

Figure 1: General System Landscape for Transportation Companies

2 Use Cases for Simulations of Pickup and Delivery

In this section, we will give two examples for the application of simulations at transportation companies. We will show ways to incorporate the simulation within existing system landscapes.

In transportation companies, there are three very basic tasks for each customer order:

(1) the acceptance of the order, including its submission to the company's software system

(2) the allocation and scheduling of a vehicle (or more), and

(3) the subsequent surveillance of the transport.

Each of these tasks is typically handled by different employees. Hence, a simulation can take over either one. In this paper, we suggest the simulation of task (1) and parts of

[1] In the context of this paper we focus on decision support systems for the 'daily business', i.e. dispatching of vehicles and orders.

task (3), i.e. the placing of transportation requests into the system and the 'driving' of associated vehicles. Extra events as traffic events or changes of customer orders can be modeled in order to increase realistic behavior of the simulation. In sum, the simulator needs to cover three main categories of events – orders, vehicles, and environment – as depicted in figure 2. Scheduling events are triggered by either human dispatchers (see section 2.1) or automatically by the recommendation engine (see section 2.2).

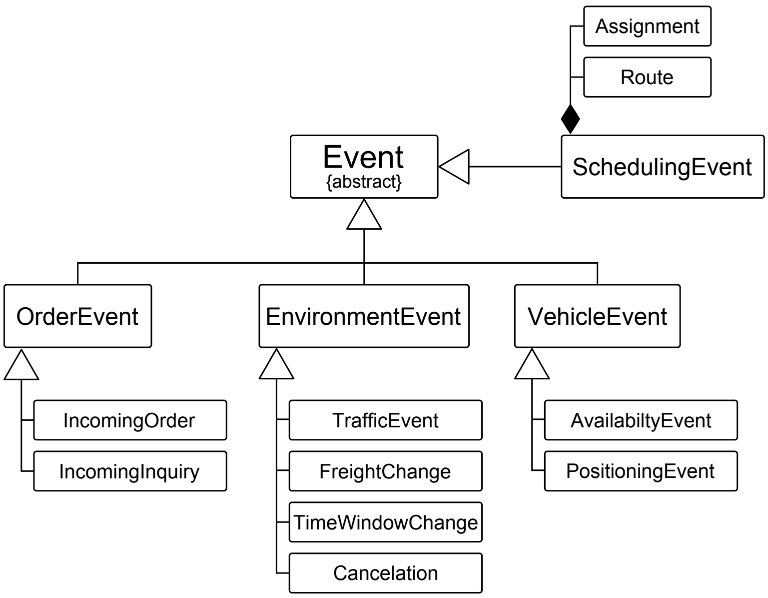

Figure 2: Categories of Events for a Simulation of Complex Pickup and Delivery Scenarios

The three categories of events presented in figure 2 are as follows:

Orders A set of transportation requests is to be released for dispatching at particular times relative to the beginning of the simulation. As we will describe in chapter 5, release timing may either be deterministic and thus repeatable or else probabilistic.

Vehicles One of the simulation's core tasks is vehicle handling. The simulator must be able to provide status and location of all vehicles at any given point in time. This challenge is further discussed in chapter 6.2.

Environment Just like customer orders, environmental events (see chapter 4) must be queued in the system. Especially cancelations can easily be treated as 'inverse orders'. Beyond that, traffic-related events like congestions, road closures or accidents must have a corresponding impact on simulated vehicles.

2.1 Education and Training

The education and training of new dispatchers in the scope of commercial transportation is closely related to the specific decision support system that is used. While the functions and display types of this software can be studied in theory, it is always best to let new users gain hands-on experience at it. Yet there is a threat on real-life systems, when unskilled users may accidentally enter wrong data and interfere with other employees' work. This can evidently be avoided by means of a simulation.

The application of internal historical data proves to be very beneficial within this context because it is already preprocessed and available in the company's database format. A simulation can simply be run on an additional database of the same format, automatically rendering it compatible with existing decision support software. That way, in task (2), human dispatchers can interact with the simulation in the exact same manner as with real-life situations; at best it will be completely transparent to them (compare figure 1 to upper part of figure 3).

2.2 Computational Tests

Another application for a simulation at a transportation company is the analysis and optimization of their decision support system; the recommendation engine in particular. For this engine there must be an underlying set of strategies used for computing allocation/scheduling suggestions. In plain theory, the system would yield acceptable transportation plans if dispatching relied completely on the recommendations. A different set of allocation/scheduling strategies for the recommendation engine can be considered superior if transportation plans that are created – again – by only following computed suggestions, are better. Quality criteria for a transportation plan include e.g. overall travel time or distance, the number of customers served, induced expenses, etc. Development and testing of new allocation/scheduling strategies can strongly be supported by a simulation. The lower part of figure 3 depicts such a benchmark scenario for a series of strategies.

Note that this use case is operated completely without human dispatchers. Therefore a computational interface is needed for exclusive interaction with the recommendation engine and the strategies that are to be examined. The interface must allow to

- assign vehicles to freight and/or vice versa,
- assign routes to vehicles,
- set departure times for vehicles,
- split freights (if allowed), and
- assign trailers to vehicles (if trailers are used).

If reloading is prohibited, vehicle-to-freight assignments also routing decisions as well, since vehicles always head directly towards pickup or delivery points, respectively. Similarly, the setting of departure times for vehicles can be interpreted as setting of waiting

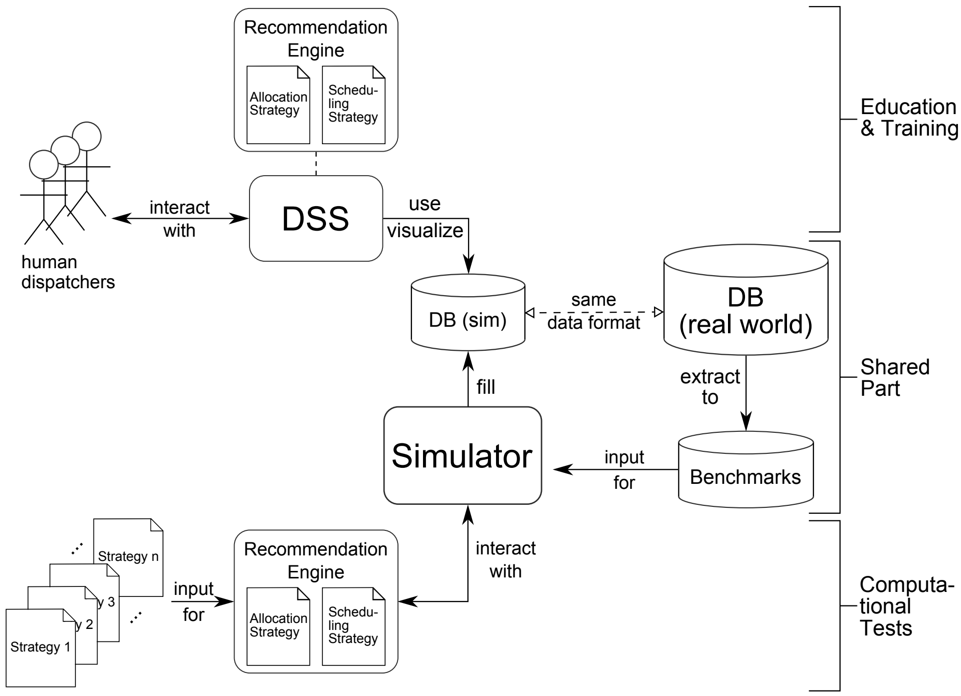

Figure 3: Simulation Setup for Transportation Companies

times. Note also that the relocation of empty vehicles may often be wise too, as addressed e.g. by [MML04].

For compatibility reasons, historical data is a good choice for this use case, too. The main difference in comparison to the educational setting is obviously possible simulation speed.

3 A Closer Look at Allocation and Scheduling Strategies

For a better understanding of allocation and scheduling strategies, in this section we will give an example of what such strategies may look like. Consider the case of a pickup and delivery service provider using a fixed set of reloading centers. Consolidation of freights from different orders can help to reduce costs and increase shipping volume of the transportation company. The following example allocation strategy aims for pairwise optimal consolidation of freights. In the course of analysis, reasonable detours to the reloading centers are considered. For this strategy, a specific data type is introduced: the *basic request*.

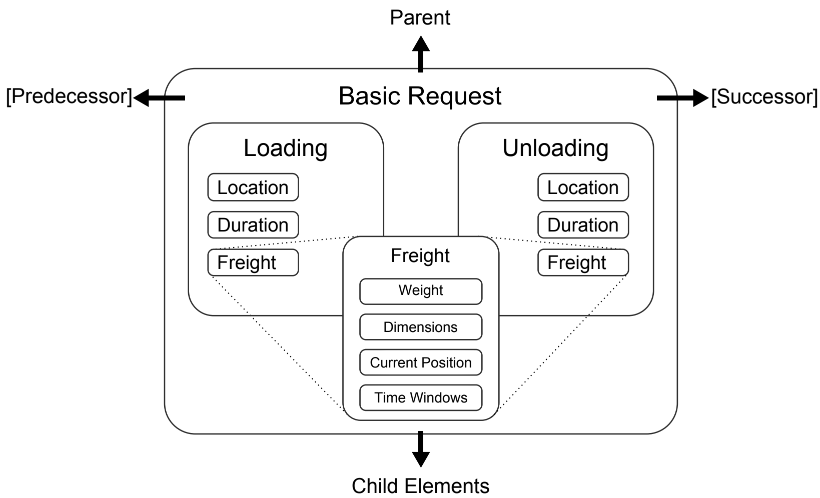

Figure 4: Structure of a Basic Request

Every basic request comprises exactly one *loading item* and one *unloading item*. These, in turn, consist of the location for (un-)loading, the estimated loading duration, and freight information including time windows. Additionally, reloading is modeled by subdivision of basic requests into exactly two closely linked child elements. The *unloading* location of the resulting first child element is always equal to the *loading* location of the second. If time windows allow for more than one reloading, subdivisions can be performed recursively leading to a tree structure as indicated in figure 4. The tree contains all possible detours to the reloading centers for the underlying customer order. Note that every basic request but the root element has either exactly one direct predecessor sibling or successsor sibling, respectively.

Evidently, every customer order can be mapped to a basic request. In a pre-processing step, the basic requests' time windows are analyzed for potential reload at the reloading centers, i.e. the 'reloading tree' for each basic request is calculated. Now, the algorithm searches for proper pairs of basic requests where the second element of the pair is part of the tree for o_{new}. The basic routine for a given o_{new} is described below.

```
1:  {Initialization}
2:  bestPair ← NULL
3:  bestMatchingValue ← −∞
4:
5:  {Main Part}
6:  for all o_old in open orders do
7:     if time windows of o_old and o_new do not overlap then
8:        goto next open order
9:     end if
10:
11:    if combined freight of o_old and o_new is too large then
12:       goto next open order
13:    end if
14:
15:    for all br_old in basic request tree for o_old do
16:       if time windows for br_old and o_new prevent consolidation then
17:          skip subtree starting from br_old
18:       end if
19:
20:       testPair ← (br_old, o_new)
21:       testValue ← calculateMatchingValue(br_old, o_new)
22:       if bestMatchingValue < testValue then
23:          bestPair ← testPair
24:          bestMatchingValue ← testValue
25:       end if
26:    end for
27: end for
28: return bestPair
```

The best pair is suggested for consolidation of freights; quality criteria (line 21) can be taken from chapter 2.2. During the search, the tree structure is exploited so that not every pairing is examined (line 17).

4 Data Sources

This section will give an overview where relevant data for simulations can be taken from. It is geared to the three categories of events, introduced in section 2.

Orders Each transportation request contains at least the exact locations for both pickup and delivery, information on the freight that is to be delivered (e.g. weight, number of items, occasionally dimensions), and – if desired – time windows. Often the timestamps for creation of the orders are saved automatically, too, by the underlying database system. Clearly, historical data of real-life customer orders can easily be incorporated into simulations.

Vehicles Today, master data on company-owned vehicles is generally already stored in companies' life system databases. Relevant data includes at least information on the vehicles' capacity and the availability of a hitch. If external vehicles are employed on a regular basis, their data is equally recorded. Depending on the size of the area served, companies also model the speed of their vehicles. This data, however, can only be used in a generalized way, due to interdependencies with assigned routes (e.g. highways, expressways, motorways, etc.), external events, and particularly local regulations on driving hours (see chapter 6.3). The mapping of an average speed to every single vehicle is normally not reasonable. Instead, a standard average speed for the whole fleet is taken as a basis. In case of a mixed fleet, transportation companies may distinguish between vehicle categories such as cars, vans, and trucks.

Furthermore, modern delivery vans and trucks are equipped not only with navigational systems but additionally with tracking devices that transmit the current position within certain intervals. This positioning data enables transportation companies to quickly respond to new customer orders, e.g. by assigning the closest vehicle available. While this may be the primary reason for a carrier to collect positioning data, it can serve for analysis and research purposes as well.

Environment While data for orders and vehicles is likely to be found in the databases of transportation companies, environmental events such as traffic jams are far less frequently recorded; many times they are completely dispensed with. An obvious reason for this lies in the extra storage space that these records would require. Thus, most life system database models are simply not designed to incorporate any data beyond the scope of primary business.

The trivial way to handle the lack of environmental data is to simply neglect it. Generally this is indeed a viable solution, as allocation and scheduling routines have to be optimized for a stable surrounding in any case. Preferably however, data analysis is conducted beforehand as a means to extract possible environmental influences. If, for example, analysis reveals regularities like seasonal fluctuation of customer orders or time-related traffic congestions, these can be incorporated into the simulation.

5 Compilation of Benchmarks

As described in chapter 1, calculations for vehicle allocation and scheduling, as discussed in this paper, rarely start from scratch. Instead, new customer orders constantly need to be incorporated into existing transportation plans. Therefore every benchmark is preceded by an initialization phase in which corresponding values are (optionally) assigned to all types of data sources.

- New customer orders are queued before the start of the simulation. For a more realistic scenario, dispatched orders that are already en-route are queued as well.
- Status and availability of vehicles are determined and vehicles are positioned on the map. In accordance to customer order initialization, some should already be assigned to a transportation request and be on their way to either a pickup or delivery point.
- Most environmental events do not require initialization. Especially cancelations and freight changes only make sense once the simulation is up and running. Traffic jams, on the other side, are set up as desired.

In the following sections different ways for data compilation are discussed. Note that these are applicable for both the initialization phase and the actual benchmarks.

5.1 Time Machine Approach

This approach may well be the most intuitive one. The idea behind this is to pick scenarios from the past and re-run them in simulation. Unlike the lessons learned approach (see below), this approach allows for a way of data mining historical decisions. If supported (see e.g. [KB01]), the underlying database system may help extracting consistent data for the benchmarks.

Advantages This approach enables re-enactment of historical data in a movie-like manner. If original allocation and scheduling decisions are replayed as well, the company's decision support software can be used in order to repeatedly examine benchmark scenarios in real-time from different 'angles'. This may, in turn, reveal strengths and weaknesses of that software with respect to the quality of visualization and/or decision supporting.

Disadvantages By nature, the time machine approach only aims to improve readiness for situations that have occurred before. Depending on the stability of transportation requests, this may not lead to appropriate benchmarks. Also, care must be taken to avoid overfitting of strategies to the benchmarks. A strategy that provides a particularly good solution for one time machine benchmark does not necessarily have to perform well in others, let alone in real-life situations.

5.2 Statistical Approach

In contrast to the time machine approach, the statistical approach generates artificial samples. Costumer orders are analyzed for frequency, timing and regional distribution. Likewise, the weight, number of items, and possibly the dimensions of freights are examined, too. Even environmental events can be modeled on such a basis. Consequently, a set of benchmarks is calculated as a mean over the analyzed samples.

Advantages Benchmarks for the statistical approach can be automatically computed. That way, test instances can easily be updated on a regular basis, allowing for quicker adaptation to changes (e.g. new key account customers) than other approaches.

Disadvantages This approach is very sensitive to the variance of the underlying samples. There is arguably no such thing as a meaningful *average order* or *average freight*. So the significance of derived benchmarks should always be subject to further analysis.

5.3 Lessons Learned Approach

In analogy to unit tests in programming (see e.g. [Ham04] or [HT03]), the this approach provides repeatable, deterministic, and discriminable test instances. In the scope of programming, the main purpose is the verification of correct behavior of software during development. In the context of a transportation company, this translates into in-depths tuning of typical or even crucial scenarios so that mistakes during the process of allocation and scheduling are not repeated.

Benchmarks can vary in numerous degrees of freedom. For the majority of pickup and delivery companies, for example, there is a significant difference concerning customer order frequency between business on weekdays and on weekends. Similarly, benchmark instances can be modeled particularly with regard to an efficient dispatch of trailers. Note that the lessons learned approach can be combined with both the time machine and the statistical approach. In contrast to the stand-alone time machine approach, however, this approach is typically used for smaller benchmarks, i.e. the duration of test instances is shorter and/or less orders and/or vehicles are incorporated.

Advantages This approach is particularly suitable if practical experience has shown that certain constellations of vehicle and order distribution repeatedly yield poor transportation plans. Thus, benchmarks can act as a continuously growing memory for common mistakes at allocation and/or scheduling. Another field of application for this approach is the separate optimization of service for certain customers such as key accounts.

Disadvantages The identification of lessons learned test instances is in itself not a trivial task. Also there is no guarantee that if allocation/scheduling strategies for these benchmarks are improved, this will increase overall business performance. If the selected test cases represent only a minority of everyday situations, tuning towards them might even have the opposite effect.

6 Limitations of Simulations

Obviously a simulation may never be a perfect substitute for reality. For a complex system like the one described in this paper, there are by far more degrees of freedom than can ever be modeled within a simulation. Hence in this section, we will take a closer look at the main limitations for a simulation of pickup and delivery scenarios.

6.1 Data Quality: Lacking and Vague Data

One of the biggest challenges when employing historical data for benchmarks is data quality. While GPS data quality for vehicle locations is more than sufficient, arguably simple data as for orders and freights is often not very accurate in praxis. This is particularly the case when customer orders are business to business. For large freights e.g. in the automotive sector, it is not unusual for one company to place procurement orders for another without knowledge of the exact weight, dimensions or number of the transported goods. As with average speed (see chapter 4), the lack of information can be coped with by means of abstraction. Freight categories can be introduced so as to provide at least some order of magnitude; preferably these correlate with vehicle categories.

Another reason for less planning reliability relates to the customers; namely their shipping processes. (Re-)Loading times can vary a lot among carriers and may account for a very high proportion of the overall transportation time. Prediction of these times is difficult as they heavily depend on the current situation at the carrier, e.g. there may be many other vehicles waiting with only one loading zone available. Modeling loading durations for each carrier separately would cause a considerable overhead. A simple way to incorporate loading times into transportation plans is to do calculations with a single *general loading duration*. It should be set long enough to cover the majority of cases. This approach will mostly overestimate loading durations and – unlike an underestimation – still lead to feasible transportation plans.

6.2 Map Matters: Distances and Locations

For every customer order, the knowledge of distances and routes between locations is paramount for planning purposes. Whether transportation companies run their own map server or use third party services, the determination of these two values remains a bottleneck. This applies especially for computational analysis scenarios such as the above mentioned recommendation engine of a decision support system. When comparing many alternative routes and plans, distance calculations can easily be the most time-consuming part. For practical reasons, a recommendation engine's heuristics may need to revert to air-line distances and only analyze the most promising results in detail.

When modeling a simulation for a transportation setting, the question of positioning for the vehicles arises, i.e. when a vehicle is on the way, where should it be localized after a certain amount of time? Surely, if routes are chosen that stem from historical data, most exact positioning is available. Yet driving of new routes must be emulated, too. As a solution, the entire simulation could be designed for air-line distances. However, these distances would continuously be underestimations. Thus average speeds, loading times and other time-related activities would need adjustments for comparison with historical routing plans in order to answer questions like 'Which routing plan is better?'. Also, the actual road network (i.e. even the *availability* of roads) would be neglected and vehicle positioning events (see figure 2) would suffer from a lack of quality.

6.3 Journey Time: Regulation of Driving Hours

The simulation of driving times is closely associated with above described map-related concerns. In reality, these times are additionally affected by regulations on driving hours.[2] While there are already mathematical models available that incorporate these regulations (see e.g. [KM09]), they still impose great challenges for both academic and practical settings. Extending the vehicle model for this purpose is not adequate since employed drivers can easily share vehicles. Thus the inclusion of regulated driving hours into the simulation can only be achieved by modeling drivers as well. However, due to the complexity of the regulations, taking rest periods in praxis is often a 'strategic' issue, as for example working periods may be prolonged only a couple of times during a week. Once more, adapted average speeds can serve as a compromise, particularly when applied to longer distances.

7 Conclusion and Outlook

In this paper, we introduced two applications for simulations in commercial transportation settings; one for the education and training of dispatchers and one for analysis and optimization of decision support software. We argued that the use of historical data can be most beneficial for transportation companies for these simulations and showed how simulations can be organized. Furthermore, we explained the idea of allocation/scheduling strategies by detail. In order to compose benchmark tests that can be run in simulation, three different approaches were presented. In the last section, we discussed the limitations of the simulation.

Currently we are working on a similar setting in order to research allocation and scheduling strategies for the general dynamic pickup and delivery problem. Having access to five years worth of real-life historical data from a large business to business transportation service provider, future work will include a feasibility study of the matter.

[2] For example, in the EU drivers are required to take breaks of at least 45 min after every 4.5 h of driving. See EC Regulation (EC) No 561/2006 for further details.

References

[Dom07] Wolfgang Domschke. *Logistik: Transport: Grundlagen, lineare Transport- und Umladeprobleme*. Oldenbourg, 5th edition, 2 2007.

[Ham04] Paul Hamill. *Unit Test Frameworks*. O'Reilly Media, Inc., 10 2004.

[HT03] Andy Hunt and Dave Thomas. *Pragmatic Unit Testing in Java with JUnit*. The Pragmatic Programmers, 1st edition, 9 2003.

[KB01] Sushil Kumar and Tammy Bednar. Oracle9i Flashback Query. White Paper, August 2001.

[KM09] Herbert Kopfer and Christoph Manuel Meyer. A Model for the Traveling Salesman Problem including the EC Regulations on Driving Hours. In *Operations Research Proceedings 2008*, pages 289–294. Springer, 2009.

[MML04] S. Mitrovic-Minc and G. Laporte. Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Res. Part B*, 38:635–655, 2004.

[Ohr08] Claudius Ohrt. *Tourenplanung im Straßengüterverkehr*. Gabler, 1st edition, 2008.

[SBS88] M M Solomon, E K Baker, and J R Schaffer. Vehicle Routing and Scheduling Problems with Time Window Constraints. In *Efficient Implementations of Solution Improvement Procedures. North-Holland: Elsevier Science Publishers B.V*, 1988.

[SS95] M. W. P. Savelsbergh and M. Sol. The General Pickup and Delivery Problem. *Transportation Science*, 29(1):17–29, February 1995.