Simulation-based evolution of municipal glass-waste collection strategies utilizing electric trucks

Stefan Vonolfen*, Michael Affenzeller, Andreas Beham, Stefan Wagner
School of Informatics, Communications and Media
Upper Austria University of Applied Sciences
Heuristic and Evolutionary Algorithms Laboratory
Softwarepark 11, 4232 Hagenberg
Austria
*stefan.vonolfen@heuristiclab.com

Efrem Lengauer
School of Management
Upper Austria University of Applied Sciences
Logistikum Research
Wehrgrabengasse 1-3, 4400 Steyr
Austria

***Abstract*—In urban areas freight transport imposes many social and environmental issues. Solid waste collection accounts for a considerable amount of freight transportation in municipal areas. We propose a simulation-based approach to evaluate different scenarios where electric trucks replace conventional trucks for the collection of municipal glass-waste. Under special consideration of the characteristics of electric trucks, the simulation of real-time fill level information is used and coupled with an optimization component to evolve adapted waste collection strategies. We illustrate our approach on two test-scenarios based on real-world data. We use the simulation model to evaluate several scenarios in terms of costs and environmental impact and the optimization environment to generate collection strategies that minimize those performance figures while maintaining a given service quality.**

***Index Terms*—Simulation-based optimization, waste management, electric trucks, inventory routing**

I. Introduction

According to data provided by the European Commission [1], 19.5% of all greenhouse gas emissions in the EU were caused by transport in the year 2007. Also, road freight transport grew by 2.9% annually between 1995 and 2008.

Especially in urban areas, increasing freight transport (and traffic in general) imposes social, environmental and logistical issues. Rodrigue and his colleagues [2] point out several challenges facing urban transportation such as traffic congestion, air pollution, greenhouse gas emissions and safety.

Previous studies have shown that solid waste collection accounts for a considerable amount of freight transportation in municipal areas. One example is the case study conducted by Johannson [3] who identified, that solid waste collection are estimated to account for 10-15% of the total freight transportation in the city of Malmoe in Sweden.

Municipal waste generation increased in almost all European countries since 1995 and in most countries the increase is at least greater than 10% according to the European Commission and Eurostat [4].

Pollution is an increasingly important issue, especially in city centers. Some early case studies using zero-emission electric garbage trucks have been carried out in Europe, for example in Huddersfield (UK) or in Courbevoie (France).

In this work, we focus on scenarios where glass-waste is collected using electric trucks in an urban environment. The collection process is optimized in terms of costs, resource usage, service quality and greenhouse gas emissions.

The optimization is done considering properties of electric trucks, such as comparatively low capacity, slow speed and limited range. Furthermore, real-time fill-level information of the glass-waste containers is modeled to account for fluctuations in the waste generation and allow a dynamic planning and optimization process. Johannson [3] points out the advantages of a dynamic model for waste-collection scenarios.

The scenario consists of several containers and a fleet of electric trucks. The goal is to collect the glass-waste in such a way, that the costs and distance are minimized while preventing the containers to be overfilled.

To account for the stochastic and dynamic nature, the scenarios are modeled and optimized using a simulation-based optimization approach. The waste-generation process is modeled using appropriate statistical distributions to create virtual sensors for the waste-collection containers. We autonomously evolve collection strategies in the simulation environment to adapt them to different scenarios especially considering the characteristics of electric trucks. The evolved collection strategies can then be applied in the continuous planning process.

Due to an infinite planning horizon long term effects have to be considered when making short-term decisions. Considering the complexity of the problem, we model the routing and collection process as a stochastic inventory routing problem (SIRP) that is solved each day considering the current fill-level information.

In the literature, there are many different solution strategies for the SIRP. For instance, Hemmelmayr and colleagues [5] use a combination of integer programming and variable neighborhood search to solve a stochastic inventory routing problem. Adelmann [6] and Berman [7] examine stochastic inventory routing problems without inventory holding costs. Adelman [6] uses a price-directed approach, whereas Berman and colleagues [7] apply stochastic dynamic programming. On the contrary, Kleywegt and colleagues [8] consider problems with inventory holding costs and use Markov Decisions Process models for their solution approach.

The rest of this paper is organized as following: In Section II we provide the problem formulation, in III we outline our methodology, in IV we introduce our test scenario, in V we present the results and in VI we summarize our findings and give an outlook.

II. Problem formulation

In this work, we model the glass-waste collection process as a stochastic inventory routing problem. Inventory routing can be applied to glass-waste collection by "‘reversing"' the original problem formulation. Instead of delivering products to customers, the different types of glass are instead collected from the containers in such a way that the containers are never overfilled. However, the structure and nature of the original problem formulation stays the same.

Many different variants of the SIRP are examined in the literature that differ in non-trivial details. Recent literature reviews are given for instance by Cordeau [9] or Bertazzi [10].

In our problem formulation, there are several locations N at which multiple containers P can be located. Each container p has a certain glass type t associated (white or stained glass) and has a given capacity C_p. The glass waste is collected by a homogeneous fleet of vehicles (M), each with a known capacity C_v. The total capacity C_v of a vehicle is split up into several sections C_{vt} where $\sum C_{vt} = C_V$ since the different glass types cannot be transported mixed in one section.

The planning process is performed on discrete time steps $t = 0, 1, ...$ which in our case are days. For each combination of day t and container p there is a probability distribution P_p^t given for the glass-waste production which does not change over time. For each container the fill level X_p^t can be measured on a daily basis.

For each day, the decision has to be made what locations to visit and what containers to empty ($d_p^t \in 0, 1$). The visited locations are combined into vehicle routes R, where each route r has a certain length L_r. A container always has to be fully emptied.

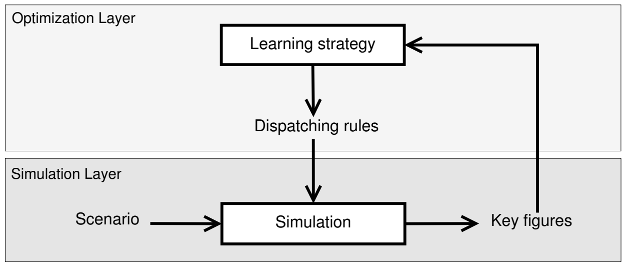

Fig. 1. System architecture.

The set of feasible solutions is determined by the vehicle capacity constraints C_v, such that

$$\sum_{d_p^t \in r} X_p^t \leq C_v$$

for all routes.

The goal is to minimize the required vehicle fleet $| N |$ and the driven distance $d = \sum L_r$ while maintaining a certain service level s, where s is determined by the number of containers over time that are overfilled (where $X_{tnp} = C_{np}$).

III. Methodology

To account for the dynamic and continuous nature of the planning process, we autonomously derive collection policies that can be adapted to diverse scenarios (e.g. to account for the situation in different urban areas). The basic system architecture to achieve that is illustrated in Figure 1.

The simulation environment receives a certain scenario as an input. A scenario consists of master data like customer locations or storage capacities and simulation parameters like demand distributions. A simulation run generates key figures like the total driven distance or the routes driven by the vehicle fleet. These figures can be used to evaluate different dispatching rules which control the waste collection strategy used in the simulation run. The dispatching rules are evolved by a learning strategy, which in turn can use the generated key figures as feedback. The learning strategy adapts the delivery rules to a certain scenario by tuning pre-defined parameters of the rules. The approach is detailed in the following.

A. Simulation environment

Core of the simulation environment is an agent-based simulation model that can be used to simulate and optimize practical transport logistic scenarios. The details on the generic model can be found in [11]. For the implementation the Repast.NET framework was used which is described by [12]. During a simulation run key figures are gathered and written to a database. Basically the model consists of three types of agents: customers, vehicles and vendors. This generic model has been adapted to the glass-waste collection problem environment.

In our model, glass-waste containers have fixed locations given in GPS coordinates. The distance matrix was created using the open route service[1] which was developed by [13]. The scenario consists of several containers and a fleet of electric trucks.

The simulation environment can be used to simulate diverse scenarios and model the characteristics e.g. of different cities across Europe. Furthermore, real-time fill-level information of the glass-waste containers is simulated to account for fluctuations in the waste generation and allow a dynamic planning and optimization process. That way virtual sensors are created that simulate the glass-waste production and can be polled every day.

B. Dispatching rules

Basically, the dispatching rules are the core of our approach and determine what containers to empty at what locations. The main objective is a constant and efficient resource utilization (e.g. vehicles, drivers) which is achieved on the one hand by minimizing the driven distance and on the other hand by maximizing the resource utilization over time while maintaining a satisfying service quality.

On the one hand, this is achieved by trying to use a constant capacity over the time and thus minimizing the required fleet by absorbing fluctuations which could be caused for example by different daily demands. This is done while preserving the service quality and satisfying the other constraints.

On the other hand, even though constant resource utilization is the main goal of our approach, in some cases it might be feasible not to perform any collection at all. That could be the case if a location is far away from the depot and the container is empty. Therefore a threshold is specified to avoid unnecessary deliveries.

As stated earlier, the dispatching rules can be parameterized to adapt them to different scenarios. There are two general parameters:

- **CapacityUtilization** The constant capacity that should be used over time is specified by this parameter.
- **PriorityThreshold** The priority threshold to be applied. If the priority of a location is lower than this level, it is not visited.

The basic algorithmic approach is outlined in pseudocode in Figure 2. First, the constructed routes (R) are initialized with the empty set. The buffer (b) is the planned capacity to be used and is initialized as a fraction of the total available capacity of all vehicles ($\sum C_v$). Then, the containers of the different locations (N) are managed by creating a set of collections (D) using two priority rules. The first priority rule (P1) is responsible for selecting a location (n) that should be visited, the second priority rule (P2) chooses the containers for each location (d_p) that should be emptied at that location. The new locations are inserted into existing routes (R) using a savings heuristic, which inserts the customer into the route r with the least detour (also considering creating a new route). After a stopping criterion has been met, the final routes are reoptimized using the push-forward insertion heuristic (PFIH) which was proposed by [14].

```
Require: $N$, $C_v$, $CapacityUtilization$, $PriorityThreshold$
 1: $R \leftarrow \emptyset$
 2: $b \leftarrow (\sum C_v) * CapacityUtilization$
 3: repeat
 4:     select $n \in N$ according to P1 with priority $p$
 5:     $N \leftarrow N \setminus n$
 6:     if $p >= PriorityThreshold$ then
 7:         $r \leftarrow SavingsHeuristic(n, R)$
 8:         $R \leftarrow R \cup r$
 9:         select deliveries $D$ for $n$ according to P2
10:         $b \leftarrow b - \sum_{\forall d_p \in D} d_p * X_p^t$
11:     end if
12: until $N == \emptyset$ OR $b == 0$
13: $R \leftarrow PushForwardInsertionHeuristic(R)$
```

Fig. 2. Delivery rule

The two priority rules (P1, P2) are detailed in the following.

1) Priority Rule 1: The first priority rule is responsible for choosing a location n out of a set of locations that yet have to be visited (N).

The main goal is ensuring a desired service quality and thus prioritizing locations where the containers are full soon while minimizing the effort to integrate the locations into existing routes.

The location with the highest priority is chosen, the priority is calculated using the following formula:

$$p_n = (\sum f_{ni} * a_i)/i$$

The parameter f_{ni} represents a certain value which should be considered in the priority calculation. All parameters are normalized in the interval $[0, 1]$. Each parameter is weighted with a factor $a_i \in [-1, 1]$. Thus, the resulting priority is in the range $[-1, 1]$. The factors a_i are part of the dispatching rule and can be tuned by the learning strategy.

The following parameters are considered in the priority calculation:

- f_{n1} **- MinFullPrediction** Prediction of the number of days when the first container will be full at a customer location n.
- f_{n2} **- AvgFullPrediction** Average number of days for the prediction for different types of glass at location n.
- f_{n3} **- LastDelivery** Last visit of customer n in days.
- f_{n4} **- ContainerSize** Total container size (C_n).
- f_{n5} **- Detour** The minimum required detour to integrate customer n into the existing routes.
- f_{n6} **- Isolation** The isolation of location n, which is specified by the average distance to all other locations on the map.

The factors a_1, a_2, a_3 and a_4 are used primarily to weight the importance of the service quality, while the parameters a_5 and a_6 are used to ensure an efficient route planning.

[1]http://www.openroutservice.org

```
Require: $t$, $r$, $n$, *RefillThreshold*, *RefillBarrier*
1: $b \leftarrow AvailableCapacity(r)$
2: while $P \neq \emptyset$ AND $b > 0$ do
3:   select $p \in P$ where $\min FullPrediction(p, n, t)$
4:   $P \leftarrow P \setminus p$
5:   if $FullPrediction(p, n, t) < RefillThreshold$ OR $X_{npt} < RefillBarrier$ then
6:     $d_p \leftarrow 1$
7:     $b \leftarrow b - d_p * X_p^t$
8:   end if
9: end while
```

Fig. 3. Second priority rule

The normalization for the parameters f_{n1}, f_{n2}, and f_{n3} is achieved according to the desired planning period (in that case one week), while f_{n4}, f_{n5} and f_{n6} are normalized according to static properties of the scenario.

2) Priority Rule 2: The second priority rule is used to determine the delivered amount d_p for each product in the inventory of the chosen customer n. The main objective is to maintain a certain service quality.

The parameters of the second priority rule are the following:

- **RefillThreshold** If a container is expected to be filled shorter than the specified threshold (in percent of the planning period) it is emptied, otherwise not.
- **RefillBarrier** If the free space in a container falls below a certain barrier (in percent) it is emptied, otherwise not.

The pseudocode of the second priority rule is illustrated in Figure 3. First the available capacity b is determined by analyzing the free capacity of the route which serves the customer considering the capacity constraints of the assigned vehicle (C_v). Then, iteratively the product with the minimum predicted days it will run out of stock is chosen. The prediction is calculated by the OOSP function, which takes into account the current stock of the product (X_{npt}) and the probability distribution P_{dnp}. If the stock of the product falls below the refill barrier or the predicted days the customer will run out of the product falls below the refill threshold, the stock is refilled to a certain level (refill factor).

C. Learning strategy

As stated in the previous section, there are ten parameters to be tuned for each scenario by the learning strategy:

- CapacityUtilization
- PriorityThreshold
- $a_i, 1 \leq i \leq 6$
- RefillThreshold
- RefillBarrier

The first two parameters influence the general delivery strategy, the next six parameters the first and the last three parameters the second priority rule. The goal is to find appropriate weights that generate good results for a certain scenario. The tuning of the parameters is a continuous evolutionary process, where different parameter combinations are tried and then evaluated over a certain period of time (e.g. one month). For the learning component an evolution strategy (ES) [15] was chosen which was designed for optimizing real-valued vectors.

TABLE I
PARAMETER SETTINGS OF THE σ-SELF-ADAPTIVE ES

Parents (μ)	1
Children (λ)	3
Maximum Generations	100
Replacement	Plus
Learning parameters (τ / τ_0)	0.4 / 0.4
Mutation	Normal ($\mu = 0$, $\sigma = \sigma_i$)

Therefore, as a representation an eleven-dimensional real vector is used. Each component of the vector is mapped to a corresponding parameter and can take values in the interval $[-1, 1]$ since all parameters are normalized. By adjusting the weights the delivery strategy can be tuned. The parameters of the ES are shown in Table I.

As an evaluation of the individuals, simulation runs were performed using the components of the individual as parameters for the delivery strategy. To evaluate the individuals the following fitness function was used:

$$\min f = \alpha * d + \beta * |N| + \gamma * s$$

The total driven distance (d), the required fleet size ($|N|$) and the number of situations where the containers are full (s) are minimized subject to the capacity constraints. To account for the stochastic nature of the simulation, multiple simulation runs are performed to evaluate a single individual and an average value of those runs is calculated.

The problem was modeled in the flexible and extensible HeuristicLab optimization environment [16] and coupled with the simulation component.

IV. TEST SCENARIO

To validate our approach, we created a hypothetical test scenario that consists of 44 container locations and 131 waste glass containers, 65 of them are white-glass containers and 66 are for stained glass. The locations have been retrieved from the website of the city of Lenoding [2]. The geographical properties of the test scenario are illustrated in figure 4. The dots represent the container locations, the rectangle is the factory where the waste-glass is processed and the triangle is a possible interim storage location.

For the simulation of the waste-glass generation we use a normal distribution at each container location. Each container has a capacity of 1500 liters. We assumed a daily fluctuation of the waste-glass generation to test the adaptivity of our approach. On Sunday no waste-glass can be thrown into the containers and also no collections are made because of noise constraints. The daily parameters of the normal distribution are listed in Table II. Basically we assumed an average daily

[2] http://www.leonding.at/index.php?id=537

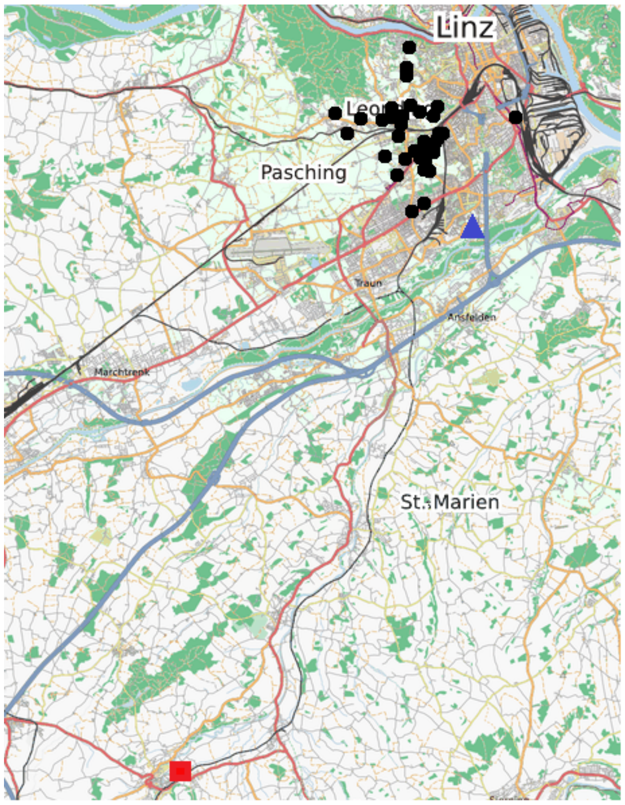

Fig. 4. Test scenario.

TABLE II
GLASS PRODUCTION DISTRIBUTION OVER THE WEEK (IN LITERS)

	Avg	Stdev
Monday	87,4015104	4,85563947
Tuesday	61,1349279	3,39638489
Wednesday	126,204416	7,01135646
Thursday	116,137369	6,45207604
Friday	124,883851	6,93799173
Saturday	114,237926	6,34655143
Sunday	0	0

waste-glass generation of 90 liters per container. The current fill level of each container can be measured using the virtual sensors of our simulation environment.

V. RESULTS

For the scenario described in the previous section we evolved a collection strategy which is listed in Table III. This parameterization has been found by the evolution strategy after multiple optimization runs and is tuned specifically for this particular scenario.

When analyzing the resulting rule, the general strategy is that containers are frequently emptied to ensure a constant resource usage. The *PriorityThreshold* parameter is rather high, which means that containers are emptied at a high rate. The high weighting of the *MinFullPrediction* and *ContainerSize* parameters prioritizes containers with critical fill levels and small containers.

TABLE III
COLLECTION STRATEGY.

CapacityUtilization	0.03
PriorityThreshold	0.68
a_1 (MinFullPrediction)	-0.50
a_2 (AvgFullPrediction)	0.00
a_3 (LastDelivery)	0.00
a_4 (ContainerSize)	0.59
a_5 (Detour)	-0.11
a_6 (Isolation)	0,05
RefillThreshold	0.41
RefillBarrier	0.80

We performed extensive simulation runs using the evolved collection strategies and tested our approach using two different strategies. For the first strategy (electric) transport the waste-glass directly to the factory using only electric vehicles and we do not use an interim storage. For the second strategy (hybrid) we additionally use an interim storage that is co-located to the containers and a conventional truck that transports the waste-glass to the factory.

The electric truck (E) has a capacity of only 2 tons and a range of a maximum of 150 km. The results show that one electric truck is sufficient in our scenario. The conventional truck (C) has a capacity of 22 tons. The simulation runs are detailed in Table IV. The results show, that the utilization of the electric truck is rather high (72,51% on average) and the waste-glass is collected constantly. This ensures a high level of service quality. When introducing an interim storage, a conventional truck picks up the waste glass every 14 days and transports it to the factory where it is processed. The driven distance of the electric truck can be reduced dramatically.

The results show that waste-glass collection using only one small electric truck is possible in our scenario. Thus conventional waste-collection trucks can be banned from city centers and replaced by small electric trucks. Of course, on the downside, the costs of that approach are expected to be higher compared when using larger trucks because the containers are emptied much more frequently and thus the traveled distance and resource usage is much higher. To mitigate that fact, larger electric trucks could be used that are expected to become available and affordable in the future.

VI. CONCLUSION AND OUTLOOK

Concluding, we have shown that it is possible to ban conventional waste-collection trucks that use combustion engines from the city center and replace them with small electric trucks that have a limited range. We have illustrated our approach on an example test scenario and presented a simulation and optimization environment to automatically derive efficient glass-waste collection strategies.

Our approach was able to produce an efficient strategy even when using a very small electric truck that is efficiently utilized. When larger electric trucks become available and

TABLE IV
SIMULATION RUNS FOR THE DIFFERENT STRATEGIES.

	Electric			**Hybrid**				
Run	Distance	Utilization	ServiceQuality	Distance (E)	Utilization (E)	Distance (C)	Utilization (C)	ServiceQuality
R1	7231.78	72.55%	100.00%	2276.63	72.94%	480.00	6.05%	100.00%
R2	7234.73	71.69%	100.00%	2285.50	71.97%	480.00	6.06%	100.00%
R3	7302.00	72.61%	100.00%	2309.00	72.28%	480.00	6.05%	100.00%
R4	7224.83	73.01%	100.00%	2293.00	73.23%	480.00	6.06%	100.00%
R5	7318.85	72.65%	100.00%	2332.83	72.84%	480.00	6.05%	100.00%
R6	7300.28	72.73%	100.00%	2367.88	73.61%	480.00	6.06%	100.00%
R7	7315.43	72.76%	100.00%	2369.33	71.72%	480.00	6.06%	100.00%
R8	7228.25	72.94%	100.00%	2294.93	71.79%	480.00	6.06%	100.00%
R9	7311.75	72.09%	100.00%	2298.00	72.35%	480.00	6.06%	100.00%
R10	7269.53	72.06%	100.00%	2346.78	72.35%	480.00	6.05%	100.00%
Avg	7273.74	72.51%	100.00%	2317.39	72.51%	480.00	6.06%	100.00%
Stdev	38.06	0.40%	0.00%	32.52	0.60%	0.00	0.00	0.00%

affordable they have potential to replace conventional trucks for tasks like waste-collection that require short-haul distances. Additionally they can be combined with conventional trucks when longer distances are required.

In the future it would be interesting to consider different scenarios that utilize electric trucks and perform case-studies where it is possibly to utilize them efficiently. For more complex environments sophisticated collection, distribution and routing rules could be evolved using genetic programming where not only parameters are tuned but also the structure is optimized. Also it would be relevant to consider the environmental impact in the optimization process and to minimize emissions by evaluating different scenarios in the simulation environment.

ACKNOWLEDGMENTS

The work described in this article was done within the Regio 13 program sponsored by the European Regional Development Fund and by Upper Austrian public funds.

REFERENCES

[1] E. Commission, *EU Energy and Transport in Figures - Statistical Pocketbook*, 2010. [Online]. Available: http://ec.europa.eu/energy/publications/statistics/statistics_en.htm

[2] J.-P. Rodrigue, C. Comtois, and B. Slack, *The Geography of Transport Systems*. Routledge, New York, 2006.

[3] O. Johansson, “The effect of dynamic scheduling and routing in a solid waste management system,” *Waste Management*, vol. 25, pp. 875–885, 2005.

[4] E. Commission and Eurostat, *Environmental Statistics and Accounts in Europe*. Publications Office of the European Union, Luxembourg, 2010.

[5] V. Hemmelmayr, K. Doerner, R. Hartl, and M. Savelsbergh, “Vendor managed inventory for environments with stochastic product usage,” *European Journal of Operational Research*, vol. 202, pp. 686–695, 2009.

[6] D. Adelman, “A price-directed approach to stochastic inventory/routing,” *Operations Research*, vol. 52, pp. 499–514, 2004.

[7] O. Berman and R. Larson, “Deliveries in an inventory/routing problem using stochastic dynamic programming,” *Transportation Science*, vol. 35, pp. 192–213, 2001.

[8] A. Kleywegt, V. Nori, and M. Savelsbergh, “The stochastic inventory routing problem with direct deliveries,” *Transportation Science*, vol. 36, pp. 94–118, 2002.

[9] J.-F. Cordeau, G. Laporte, M. Savelsbergh, and D. Vigo, “Vehicle routing,” in *Handbooks in Operations Research and Management Science*, C. Barnhart and G. Laporte, Eds. Elsevier, North-Holland, Amsterdam, 2007, vol. 14, pp. 367–428.

[10] L. Bertazzi, M. Savelsbergh, and M. Speranza, “Inventory routing,” in *The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces Serices*, B. Golden, S. Raghavan, and E. Wasil, Eds. Springer, 2007, vol. 43, pp. 49–72.

[11] S. Vonolfen, S. Wagner, A. Beham, M. Kofler, M. Affenzeller, E. Lengauer, and M. Scheucher, “A simulation-based approach to the vehicle routing problem,” in *22nd European Modeling and Simulation Symposium EMSS 2010, Fes, Marokko*, 2010, pp. 363–368.

[12] J. Vos, “The repast framework implemented in the .net framework,” in *Proceedings of the 2005 NAACSOS Conference. June 26-28, Notre Dame (Indiana, USA).*, 2005, pp. 363–368.

[13] P. Neis and A. Zipf, “Zur kopplung von opensource, openls und optenstreetmaps in openrouteservice.org,” in *AGIT 2008. Symposium fr angewandte Geoinformatik. Salzburg. Austria.*, 2008.

[14] S. Thangiah, “A hybrid genetic algorithms, simulated annealing and tabu search heuristic for vehicle routing problems with time windows,” *Practical Handbook of Genetic Algorithms, Volume II: Complex Structures*, pp. 347–381, 1999.

[15] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies - A comprehensive introduction,” *Natural Computing*, vol. 1, no. 1, pp. 3–52, March 2002.

[16] S. Wagner, “Heuristic optimization software systems - Modeling of heuristic optimization algorithms in the HeuristicLab software environment,” Ph.D. dissertation, Johannes Kepler University, Linz, Austria, 2009.