MODELING RICH AND DYNAMIC VEHICLE ROUTING PROBLEMS IN HEURISTICLAB

Stefan Vonolfen[(a)], Michael Affenzeller[(b)], Stefan Wagner[(c)]

[(a) (b) (c)] Upper Austria University of Applied Sciences, Campus Hagenberg
School of Informatics, Communication and Media
Heuristic and Evolutionary Algorithms Laboratory
Softwarepark 11, A-4232 Hagenberg, Austria

[(a)]stefan.vonolfen@heuristiclab.com, [(b)]michael.affenzeller@heuristiclab.com, [(c)]stefan.wagner@heuristiclab.com

ABSTRACT

Transport is one of the largest emission driving forces and has many economic and social impacts. Thus it is crucial to model and optimize practical transportation problems. In this paper we present a flexible modeling and optimization framework integrated in the open source environment HeuristicLab. We show, how rich and dynamic vehicle routing problem variants can be integrated in our framework. Using this model, we perform an algorithmic study where we compare several heuristic and metaheuristic algorithms for the dynamic pickup and delivery problem with time windows.

Keywords: dynamic vehicle routing problem, simulation, optimization

1. INTRODUCTION

According to a recent report by the European Commission and Eurostat (2011), transport caused 19.5% of all greenhouse gas emissions in 2008. The emissions generated by transport grew by 5% between 1999 and 2008 and thus transport is one of the largest emission driving forces. This is explained by increasing transport volumes and a lack of cleaner fuels and modes. As a result, an energy-efficient transport is an important contribution to a sustainable development.

The vehicle routing problem (VRP) is an important problem class in operations research (OR) since it can be used to model many different types of transportation problems. Since its original formulation by Dantzig and Ramser (1959), many variants have emerged and have been successfully applied in practice. For a taxonomic overview of different VRP variants the reader is referred to Eksioglu et al. (2009). A survey on recent advances and challenges in the field of vehicle routing is given by Golden and his colleagues (2008).

Challenges in contemporary vehicle routing research are on the one hand rich models that include many practical side constraints (Hartl et al. 2006) and on the other hand dynamic and stochastic formulations (Zeimpekis et al. 2007, Pillac et al. 2011).

Thus it is crucial to have a flexible software platform that can be applied to various variants of dynamic vehicle routing problems. In this work, we present how rich, dynamic vehicle routing problems are modeled in the open-source platform HeuristicLab (Wagner, 2009).

2. METHODOLOGY

To model dynamic VRPs, a simulation component is required that replaces the real practical environment. It can be used both for algorithm development and scenario evaluation. Thus, the combination of simulation and optimization is a powerful technique in that context.

The overall model is illustrated in Figure 1. The simulation component contains a problem environment that specifies the constraints such as number of vehicles, capacities or the underlying transport network. The simulation model is based on Vonolfen et al. (2010) and has been adapted to dynamic problems.

The problem environment generates orders dynamically which are transportation requests that have to be served by a fleet of vehicles. The vehicles act inside the problem environment and deliver the orders given the constraints.

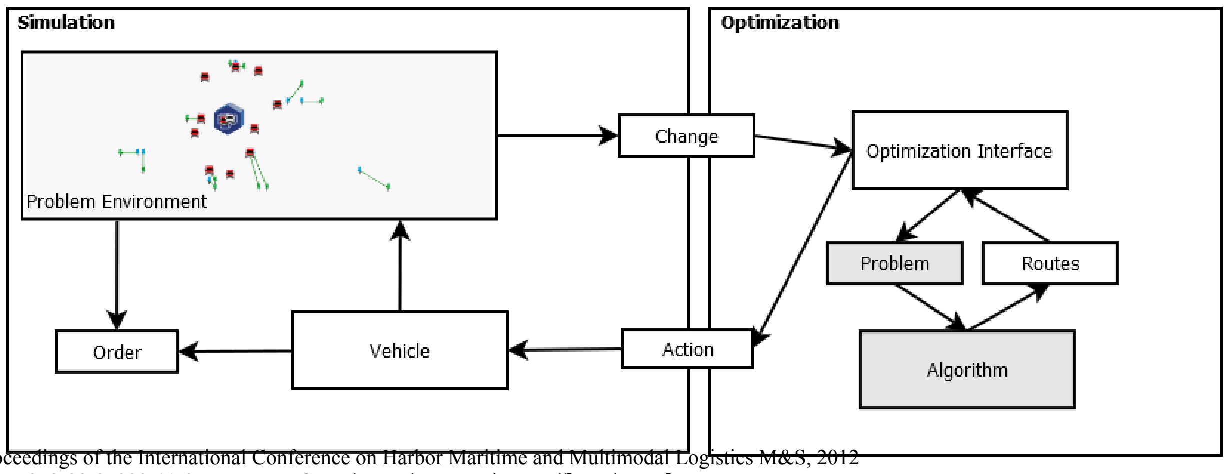

Figure 1: Simulation and Optimization

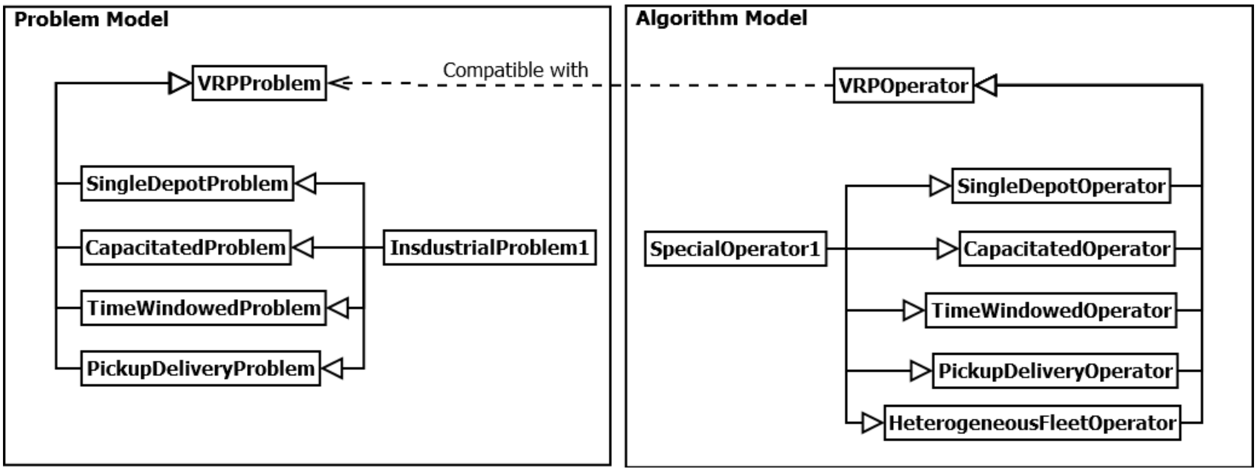

Figure 2: Problem and Algorithm Model

Whenever a dynamic event happens in the problem environment (e.g. an order is delivered, a new order appears or a vehicle breaks down), a change event is triggered and the optimization component is notified. The optimization component reacts to changes and triggers actions that are performed by the vehicles.

Different routing algorithms implemented in HeuristicLab are integrated by the optimization interface. The optimization interface transforms the current situation (including the recent changes) to a problem model and calls the underlying algorithm. The algorithm returns a route plan which is mapped to actions that are performed by the vehicles.

The grey components in the picture (problem and algorithm) have to be adapted to the problem environment and are highly dependent on the side constraints. Thus, a flexible problem and algorithm model is required in HeuristicLab to be able to model diverse rich variants of practical VRPs.

The model is illustrated in Figure 2. Each problem model requires certain side constraints that have to be considered by the algorithm solving it. For instance, practical problems may consist of multiple depots, a heterogeneous fleet or incorporate pickup and delivery operations.

The algorithm is composed of building blocks called operators. The operators offer certain features that can be considered. The algorithm can be modeled in a flexible way. For a certain problem environment, suitable operators can be chosen that can consider the required side constraints. Operators developed for other rich practical VRP variants can be reused in other problem environments.

3. PROBLEM FORMULATION

To illustrate our model, we implemented an example practical VRP variant and tested different algorithms for that problem environment.

In particular, we consider the dynamic pickup and delivery problem with time windows (PDPTW) which can be considered as a rich transportation problem. Practical applications of the dynamic PDPTW are manifold and include full-truckload problems, less-than-truckload problems and passenger transportation.

Its formulation is based on the static PDPTW model (Savelsbergh 1995). A fleet of vehicles has to serve a set transportation requests during a planning period (i.e. a day of operation). Each request is characterized by a pickup and a delivery location and the size of the load to be transported. For each location, a time window is given in which the service has to occur. A request has to be fulfilled by exactly one service of a single vehicle; this means that split deliveries are not allowed. In the dynamic formulation, not all requests are known in advance but are revealed during the planning period.

Dynamic vehicle routing problems are characterized by changing information and the routes evolve regarding to those inputs in real-time (Psaraftis 1988). The focus will be on the arrival of new requests during the planning process. At a certain time instant *t* the route plan is divided into three parts (Ichoua et al. 2007): the completed movements, the current movement and the planned movements.

Early research on dynamic PDP includes Shen et al. (1995) and Potvin et al. (1995) where they apply neural networks with learning capabilities.

In terms of neighborhood based metaheuristics, a tabu search algorithm with a neighborhood elimination matrix was applied by Malca and Semet (2004). Gendreau and Potvin (1998) apply a tabu-search heuristic based on a neighborhood of ejection chains. A two-phase solution approach where a tabu search is combined with different waiting strategies is examined by Mitrovic-Minic and Laporte (2004).

But also population-based metaheuristcs have been applied successfully. A grouping-based genetic algorithm is applied to a set of benchmark instances by Pankratz (2005). A genetic algorithm hybridized with fuzzy clustering for predictive control is presented by Saez et al. (2009).

4. ALGORITHMS

In this work, two different strategies for optimizing dynamic pickup and delivery problems are examined, namely updating of the current plan and complete reoptimization. The first approach corresponds to integrating the new requests in the planned movement while the second approach relies on a complete reoptimization whenever new information is revealed. Both commonly used heuristics as well as metaheuristics are analyzed. The algorithms are detailed in the following.

4.1. Heuristics

A straightforward approach of updating the current route plan according to newly arriving requests is to insert them at the best possible position of the planned routes. This is often referred to as the best insertion heuristic. A major drawback is that decisions made in the past that correspond to planned movements cannot be changed at a later time during the planning process.

This issue can be overcome by a complete reoptimization of the planned routes at each time step *t* given the current situation. This can be done for instance by means of a construction heuristics. A push forward insertion heuristic is examined which has been originally proposed by Solomon (1987) for the vehicle routing problem with time windows (VRPTW). It has been adapted to the PDPTW by Li and Lim (2001). It basically inserts pairs of locations into routes. A pair of locations consists of a pickup and a delivery location. First a route is initialized with a pair based on the distance to depot and time windows. Then iteratively a pair that causes the minimum insertion costs is inserted into the route until no pair can be feasibly inserted. Then a new tour is started. This procedure is repeated until all pairs of customers are routed.

4.2. Metaheuristics

Two metaheuristics for the optimization of dynamic pickup and delivery problems are compared, namely a genetic algorithm and tabu search. Those two algorithms have been used frequently for the PDPTW in the literature. In terms of parameter setting, a single set of parameters has been tested which has been determined empirically.

The applied genetic algorithm uses mutation and crossover operators proposed by Potvin and Bengio (1996); i.e. the one-level exchange mutator, two-level exchange mutator, route-based crossover and sequence-based crossover. They are implemented using a route-based encoding which is examined and compared with other encodings by Vonolfen et al. (2012). The initial generation is obtained by using the before mentioned push forward insertion heuristic. In terms of algorithm parameters, a population size of 50 is used with a tournament selection and a mutation probability of 5%. Whenever a new request arrives, the algorithm is given 100 generations to compute a new route plan.

The tabu search algorithm utilizes three different neighborhoods that have been proposed by Li and Lim (2001). The shift neighborhood considers moves where pickup and delivery customer pairs are shifted from one route to another. In the exchange neighborhood pairs are swapped between two routes. Within one route pairs can be moved to another position in the rearrange neighborhood. In each iteration, 1000 possible moves are sampled from the neighborhoods. To achieve that, 10 different neighborhoods and from each neighborhood 100 moves are sampled uniformly. As a tabu criterion, a customer cannot be moved back to a route once it has been removed or rearranged in it. A tabu tenure of 10 is used with a soft aspiration criterion for improving moves. The soft aspiration criterion accepts both new best solutions and moves that are better than the individual where the move has been set tabu. A fixed tabu tenure has been used because all tested instances consist of the same number of customers. At the arrival of a new request, 100 iterations are performed.

Whenever a new request arrives, there are two approaches in integrating it in the current route plan (Ichoua et al. 2007). The first approach is a complete reoptimization, the second approach is a local update whenever new information is revealed. In the local update approach, the route plan is not computed from scratch but information about previous time steps is used. In the case of the genetic algorithm, whenever a new request arrives it is inserted into each individual of the population using the best insertion heuristic. For the local update tabu search algorithm, the current solution and the tabu list are updated.

5. EXPERIMENTS

To evaluate the performance of the algorithms, test runs have been performed on various benchmark instances. The goal is to minimize both the required fleet and the driven distance. The test instances have been retrieved from the benchmark data set proposed by Pankratz (2005). It includes several different dynamic PDPTW instances with various properties. This means that all requests are dynamic and occur during the planning process.

The test set contains different instance types, which are the C1, C2, R1, R2, RC1 and RC2 types with different urgency factors. This sums up to a total of 12 instance classes. The C instances contain geographically clustered, the R instances randomly distributed and the RC instances both clustered and randomly distributed customers. The instances with a "1" as a suffix contain customers with large time windows as opposed to the instances with a "2" which are characterized by tight time windows. Each class contains 8-12 different instances.

All instances are based on the well-known Solomon benchmark set (http://web.cba.neu.edu/~msolomon/problems.htm). All instances consist of 100 customers making up a total of 50 dynamic requests. The best known results for the offline instances are listed in Table 1 and have been retrieved from the SINTEF website (http://www.sintef.no/Projectweb/TOP/Problems/VRPTW/Solomon-benchmark/100-customers).

Problem	Vehicles	Distance
C1	10,00	828,38
C2	3,00	589,86
R1	11,92	1209,89
R2	2,73	951,02
RC1	11,50	1384,16
RC2	3,25	1119,17

Table 1: Best Known Results for the Offline Instances

Based on these static instances, the test set contains different instance classes in terms of degree of dynamism. For the purpose of this paper, instance classes with no a-priori knowledge have been chosen. The instances have different urgency factors, namely 10% and 90%. The urgency factor determines how long in advance the request is known in relation to the latest service time and thus how long the reaction time is. The heterogeneity of the test instances allows us an algorithm performance analysis under different dynamic, spatial and temporal properties.

The dynamically arriving customers are determined beforehand and the same instances are used for all algorithms. Multiple runs are required to capture the stochastic behavior of the algorithms. For the best insertion heuristic only one single run was required, since it is deterministic.

For all other algorithms, three independent test runs have been performed on each instance and the average results in terms of fleet size (vehicles) and distance are listed.

The examined algorithms include both heuristics (best insertion, push forward insertion) and metaheuristics (reoptimization genetic algorithm, local update genetic algorithm, reoptimization tabu search, local update tabu search). They are described in the previous Section.

		Best Insertion		PF Insertion	
Urg.	Class	Vehicles	Distance	Vehicles	Distance
10%	C1	11.44	1618.88	12.56	1308.14
	C2	4	1391.61	4.63	1209.47
	R1	16.33	1785.75	17.33	1704.87
	R2	4.3	1779.31	4.7	1755.94
	RC1	17.5	2199.07	17.88	2017.66
	RC2	5.13	2149.45	6	2246.08
90%	C1	20.22	1926.49	19.78	1682.96
	C2	9.5	2291.86	9.63	1992.82
	R1	24.81	2160.19	25.27	2126.19
	R2	9.83	2508.69	10.38	2280.97
	RC1	27	2688.08	27.25	2631.74
	RC2	10.13	2818.12	10.75	2692.01

Table 2: Results for the Heuristics

The results achieved by using heuristics are summarized in Table 2. The push forward insertion heuristic generally outperforms the best insertion heuristic in terms of distance.

This can be explained by the fact that reoptimizing the routes in each time step leaves a larger room for optimization potential than gradually inserting newly arriving requests where existing plans cannot be changed. However, the push forward insertion heuristic uses a larger fleet size on average, which indicates that the parameters could be tuned to use fewer vehicles.

The results achieved by the metaheuristics are listed in Table 3. The local update tabu search is the best performing algorithm both in terms of fleet size and in terms of distance. This indicates that it can adapt the existing plan efficiently to newly arriving requests.

		Reopt. GA		Local Update GA		Reopt. TS		Local Update TS	
Urg.	Class	Vehicles	Distance	Vehicles	Distance	Vehicles	Distance	Vehicles	Distance
10%	C1	10.78	960.61	10.52	1113.3	11.33	938.93	10.52	962.44
	C2	3.92	712.73	3.67	746.14	4.08	774.04	3.75	718.87
	R1	15	1465.63	14.28	1474.89	16.19	1445.82	14.22	1404.11
	R2	3.87	1329.18	3.6	1266.52	4.9	1277.67	3.33	1160.53
	RC1	16.13	1762.6	14.92	1763.5	16.96	1735.95	14.13	1643.9
	RC2	4.67	1530.99	4.04	1463.45	5.25	1500.95	4.08	1350.67
90%	C1	19.44	1672.09	20.19	1756.41	19.48	1658.73	19.44	1638.31
	C2	8.58	1514.82	9.42	1591.36	8.63	1490.47	8.63	1528.09
	R1	23.92	2068.52	24.32	2090.12	23.46	2036.24	23.57	2036.08
	R2	8.86	2036.18	9.59	2107.94	8.55	2000.1	8.62	2020.9
	RC1	26.67	2602.16	26.71	2601.37	26.25	2566.1	26.25	2562.46
	RC2	9.25	2380.12	9.92	2479.97	9.33	2374.48	9.25	2362.02

Table 3: Results for the Metaheuristics

A summary of the conducted experiments is illustrated in Figure 3. All metaheuristic algorithms can outperform the heuristics significantly both in terms of distance and in terms of fleet size. However, it can also be observed that the performance of the algorithms decreases significantly with increasing urgency.

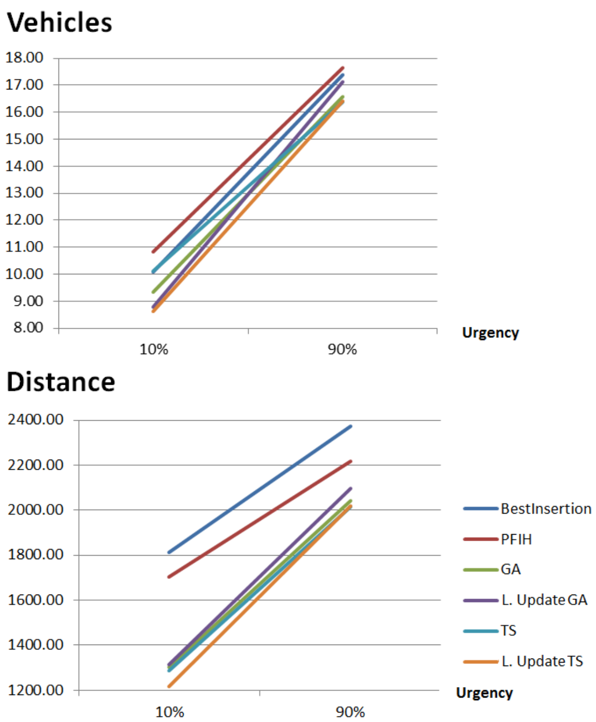

Figure 3: Summary of the Results

This shows that adapting well performing static algorithms is not sufficient for highly dynamic problems. The algorithmic concepts have to be extended by a mechanism to anticipate future requests such as waiting strategies with a double horizon approach (Mitrovic-Minic and Laporte 2004) or a scenario based technique (Van Hentenryck and Bent 2009). However, according to Berbeglia et al. (2010), the literature is still very scarce in terms of dynamic and stochastic PDP.

Work on dynamic and stochastic vehicle routing problems includes Van Hentenryck and Bent (2004), Hvattum et al. (2006) and Secomandi and Margot

(2009). The most promising approach seems to be the multi-plan approach of Van Hentenryck and Bent (2004) which shows large improvements compared to approaches that do not use information regarding customer order probabilities.

6. CONCLUSION

We have presented a flexible and extensible model for rich and dynamic vehicle routing problems. The model is incorporated in the open-source optimization framework HeuristicLab. To illustrate our approach, we have implemented an example dynamic transportation problem and evaluated the performance of different optimization algorithms.

In future work, we want to extend our problem models with stochastic aspects to anticipate future requests. Possible extensions include waiting strategies, multi-plan approaches and double-horizon approaches.

Also, the framework will be used in practical projects involving company partners to model rich transportation problems and to adapt the algorithms to the specific environments.

ACKNOWLEDGMENTS

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

REFERENCES

G. Berbeglia, J.F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. European Journal of Operational Research, 202(1):8–15, 2010.

R.W. Bent and P. Van Hentenryck. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research, pages 977–987, 2004.

G. B. Dantzig and J. H. Ramser. The truck dispatching problem. Management Science, 6(1):80–91, 1959.

B. Eksioglu, A. Volkan Vural, and A. Reisman. The vehicle routing problem: A taxonomic review. Computers & Industrial Engineering, 57(4):1472 – 1483, 2009.

European Commission and Eurostat, Viktoria Bolla, Velina Pendolovska, 2011. *Driving forces behind EU-27 greenhouse gas emissions over the decade 1999-2008*. ISSN 1977-0316. Available from: http://epp.eurostat.ec.europa.eu/portal/page/portal/product_details/publication?p_product_code=KS-SF-11-010 [Accessed 11th of April 2012].

M. Gendreau and J.-Y. Potvin. Dynamic vehicle routing and dispatching. In G. Laporte T.G. Crainic, editor, Fleet Management and Logistics. Kluwer, Boston, 1998.

B. Golden, S. Raghavan, and E. Wasil. The vehicle routing problem: latest advances and new challenges. Operations Research/Computer Science Interfaces. Springer, Dordrecht, 2008.

R. Hartl, G. Hasle and G. Janssens. Special issue on Rich Vehicle Routing Problems. Central European Journal of Operations Research, 2006, vol. 14, issue 2.

P. Van Hentenryck and R. Bent. Online stochastic combinatorial optimization. The MIT Press, 2009.

L.M. Hvattum, A. Løkketangen, and G. Laporte. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transportation Science, 40(4):421–438, 2006.

S. Ichoua, M. Gendreau, and J.Y. Potvin. Planned route optimization for real-time vehicle routing. Dynamic Fleet Management, pages 1–18, 2007.

H. Li and A. Lim. A metaheuristic for the pickup and delivery problem with time windows. In Tools with Artifcial Intelligence, Proceedings of the 13th International Conference on, pages 160–167. IEEE, 2001.

F. Malca and F. Semet. A tabu search algorithm for a dynamic pickup and delivery vehicle routing problem. In Triennal Symposium on Transportation Analysis, Le Gosier, Guadeloupe, France, juin, 2004.

S. Mitrovic-Minic and G. Laporte. Waiting strategies for the dynamic pickup and delivery problem with time windows. Transportation Research Part B: Methodological, 38(7):635–655, 2004.

G. Pankratz. Dynamic vehicle routing by means of a genetic algorithm.International Journal of Physical Distribution & Logistics Management, 35(5):362–383, 2005.

V. Pillac, M. Gendreau, C. Gueret, and A. Medaglia. A review of dynamic vehicle routing problems. Technical Report CIRRELT-2011-62, CIRRELT, 2011.

J. Potvin, Y. Shen, G. Dufour, and J. Rousseau. Learning techniques for an expert vehicle dispatching system. Expert Systems with Applications, 8(1):101–109, 1995.

J. Potvin and S. Bengio. The vehicle routing problem with time windows -. part ii: Genetic search. INFORMS Journal on Computing, 8:165–172, 1996.

H.N. Psaraftis. Dynamic vehicle routing problems. In Vehicle Routing: Methods and Studies, pages 223–249. Elsevier Science Publishers, 1988.

D. Saez, C.E. Cortes, and A. Nunez. Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. Computers & Operations Research, 35(11):3412–3438, 2008.

M. W. P. Savelsbergh The general pickup and delivery problem. Transportation Science, 29(1):17–29, 1995.

N. Secomandi and F. Margot. Reoptimization approaches for the vehicle-routing problem with stochastic demands. Operations research, 57(1):214–230, 2009.

Y. Shen, J.Y. Potvin, J.M. Rousseau, and S. Roy. A computer assistant for vehicle dispatching with learning capabilities. Annals of Operations Research, 61(1):189–211, 1995.

M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res., 35:254–265, April 1987.

S. Vonolfen, S. Wagner, A. Beham, M. Kofler, M. Affenzeller, E. Lengauer, M. Scheucher, 2010. A Simulation-Based Appraoch to the Vehicle Routing Problem. 22nd European Modeling and Simulation Symposium EMSS 2010, Fes, Marokko, 363-368.

S. Vonolfen, A. Beham, M. Affenzeller, S. Wagner, and A. Mayr. Combination and comparison of different genetic encodings for the vehicle routing problem. Computer Aided Systems Theory{EUROCAST 2011, pages 327–334, 2012.

S. Wagner, 2009. Heuristic optimization software systems – Modeling of heuristic optimization algorithmsin the HeuristicLab software environment. Thesis (PhD). Johannes Kepler Univiersity, Linz, Austria.

V. Zeimpekis, C.D. Tarantilis, G.M. Giaglis, and I. Minis. Dynamic Fleet Management. Operations Research/Computer Science Interfaces. Springer, New York, 2007.

AUTHORS BIOGRAPHY

STEFAN VONOLFEN studied Software Engineering at the Upper Austrian University of Applied Sciences, Campus Hagenberg and received his MSc in engineering in 2010. Since January 2010 he works at the Research Center Hagenberg of the Upper Austrian University of Applied Sciences. His research interests include transport logistics optimization and simulation-based optimization.

MICHAEL AFFENZELLER has published several papers, journal articles and books dealing with theoretical and practical aspects of evolutionary computation, genetic algorithms, and meta-heuristics in general. In 2001 he received his PhD in engineering sciences and in 2004 he received his habilitation in applied systems engineering, both from the Johannes Kepler University of Linz, Austria. Michael Affenzeller is professor at the Upper Austria University of Applied Sciences, Campus Hagenberg, and head of the Josef Ressel Center *Heureka!* at Hagenberg.

STEFAN WAGNER received his MSc in computer science in 2004 and his PhD in engineering sciences in 2009, both from Johannes Kepler University (JKU) Linz, Austria; he is professor at the Upper Austrian University of Applied Sciences (Campus Hagenberg). Dr. Wagner's research interests include evolutionary computation and heuristic optimization, theory and application of genetic algorithms, machine learning and software development.

More information about HeuristicLab and the research group HEAL can be obtained from http://heal.heuristiclab.com