Evolutionary Freight Transportation Planning

Thomas Weise[1], Alexander Podlich[2], Kai Reinhard[2], Christian Gorldt[3], and Kurt Geihs[1]

[1] Distributed Systems Group, University of Kassel, 34121 Kassel, Germany
[2] Micromata GmbH Kassel, Marie-Calm-Strae 3, 34131 Kassel, Germany
[3] BIBA – Bremer Institut für Produktion und Logistik GmbH

Abstract. In this paper, we present the freight transportation planning component of the *INWEST* project. This system utilizes an evolutionary algorithm with intelligent search operations in order to achieve a high utilization of resources and a minimization of the distance travelled by freight carriers in real-world scenarios. We test our planner rigorously with real-world data and obtain substantial improvements when compared to the original freight plans. Additionally, different settings for the evolutionary algorithm are studied with further experiments and their utility is verified with statistical tests.

1 Introduction

With the steadily increasing freight traffic resulting from trade inside the European Union and global import and export as well [2], the need for intelligent solutions for the strategic planning of logistics is growing steadily. Such a planning process has the goal of

1. increasing the profit by
2. ensuring the on-time collection and delivery of all parcels,
3. utilizing all available means of transportation (rail, trucks) efficiently, i. e., reducing the total transportation distances by using the capacity of the vehicles to the full, while
4. reducing the CO_2 production in order to become more environment-friendly.

Obviously, the last point is a side-effect of the others. By reducing the total distance covered and by transporting a larger fraction of the freight by (inexpensive) trains, not only the hours of work of the drivers and the costs are reduced, but also the CO_2 production.

Additionally, an efficient freight planning process has two other constraints. First, it must be dynamic and able to react to traffic jams by re-routing freight vehicles. Second, experience has shown that hiring external carriers for a small fraction of the freight in local areas can often reduce the number of required own tours significantly and thus, increase the relative utilization of the own means of transportation (see, for instance, Fig. 3.2). Therefore, freight planning is a multi-objective optimization problem and the human operators must also be provided with partial solutions to select from.

In this paper, we present the freight transportation planning component of *INWEST*, a joint project funded by the German Federal Ministry of Economics and Technology, which fulfills all these requirements. In the following section, we discuss the general requirements of the logistics departments of the project partners which specify the framework for our freight planning component.[4] These specific conditions ruled the related approaches outlined in Section 3 infeasible. In Section 4, we present freight planning as a multi-objective evolutionary optimization [3] problem. The problem-specific representation of the solution candidates and the intelligent search operators working on them are introduced, as well as the objective functions derived from the requirements already stated in this section. Our approach has been tested in many different scenarios and the experimental results are summarized in Section 5. The paper concludes with a discussion of the results and future work in Section 6.

2 Model based on the Real-World Situation

The basic unit of freight considered in this work is a swap body b, a standardized container (C 745, EN 284 [4]) with an extent of roughly $7.5\text{m} \times 2.6\text{m} \times 2.7\text{m}$ and special appliances for easy exchange between transportation vehicles or railway carriages. Logistics companies like the *DHL* own thousands of such swap bodies and we refer to their union as set B.

The set of all possible means of transportation will be referred to as F in the following. All trucks $tr \in F$ can carry $\hat{v}(tr) = 2$ such swap bodies at once whereas the capacity limits of trains $z \in F$ are usually somewhere between 30 and 60 ($\hat{v}(z) \in [30..60]$). Trains have fixed routes, departure, and arrival times. Freight trucks can move freely on the map, but must usually perform cyclic tours, i.e., return to their point of departure by the end of the day, so that the human drivers are able to return home.

The clients and the depots of the logistics companies together form roughly 1000 locations from which freight may be collect or to which it may be delivered. We will refer to the set of all these locations as L. Each transportation order has a fixed time window $[\widecheck{t_s}, \widehat{t_s}]$ in which it must be collected from its source $l_s \in L$ and a destination location and time window $[\widecheck{t_d}, \widehat{t_d}]$ in which it must be delivered to its destination $l_d \in L$. It furthermore has a volume v which is an integer multiple of the capacity of a swap body. Hence, a transportation order o can fully be described by the tuple $o = \left\langle l_s, l_d, [\widecheck{t_s}, \widehat{t_s}], [\widecheck{t_d}, \widehat{t_d}], v \right\rangle$. Depending on

[4] That partners in the project *INWEST* (Intelligente Wechselbrücksteuerung, funded by the German Federation) are the *Deutsche Post AG*, *DHL*, the *Micromata GmbH*, *BIBA*, and *OHB Teledata GmbH*; see `http://www.inwest.org/` [accessed 2008-10-29].

the day of week and national holidays etc., between 100 and 3000 such orders have to be processed per day. In the following, all orders which require more than one ($v > 1$) swap body will be split up into multiple orders requiring one swap body ($v = 1$) each.

The result of the planning process is a *set* R of tours. Each single tour r is described by a tuple $r = \langle l_s, l_d, f, \check{t}, \hat{t}, \underline{b}, \underline{o} \rangle$ where l_s and l_d are the start and destination locations, $\check{t}$ and $\hat{t}$ are the departure and arrival time, $\underline{b} = \{b_1, b_2, \dots\}$ is a set of swap bodies which are carried by the vehicle $f \in F$ and contain the goods assigned to the orders $\underline{o} = \{o_1, o_2, \dots\}$.

Additional to the constraints mentioned in the introduction, a freight planning system must ensure that the tours computed are *physically sound*. Neither freight vehicles nor orders or swap bodies can be part of more than one tour at a time and the capacity limits of all involved means of transportation must be respected. If some of the freight is carried by trains, the fixed halting locations of the trains as well as their assigned departure and arrival times must be considered. Of same importance are the restrictions imposed on the runtime of the optimization process which must not exceed one day.

3 Related Work

The approaches discussed in the literature on freight transportation planning can roughly be divided into two basic families: deterministic and stochastic or metaheuristic methods. Especially the metaheuristic optimization methods received more and more attention during the past years. The quality of solutions produced by them is often much higher than that obtained by classical heuristics.

Well-known approaches for different types of vehicle routing problems and freight transportation planning are Tabu Search [5–8], Simulated Annealing [9, 10], ant systems [11, 12], and especially evolutionary algorithms [13–16]. Most of these publications outline special types of vehicle routing problems (e.g., the vehicle routing with time windows or vehicle routing with back hauls). Usually, these problems are solved with single-objective optimization enriched with problem-specific constraints. The size of such problems is typically roughly around 100 customers or orders, as is the case in most of the example data sets in [17, 18].

The authors of the publications mentioned in the text above often indicate that metaheuristics succeed only when a good deal of domain knowledge is incorporated. This holds not only for vehicle routing, but is the case in virtually every application of global optimization [19, 20]. Nevertheless, such knowledge is generally used as an extension, as a method to tweak generic operators and methods. In this work, we have placed problem-specific knowledge in the center of the approach.

4 Evolutionary Approach

Evolutionary algorithms are a family of nature-inspired optimization algorithms which utilize natural processes such as selection and reproduction in order to refine a set (population) of solution candidates iteratively [21, 22]. This cycle starts with the evaluation of the objective values of the solution candidates. Based on these results, a relative fitness is assigned to each solution candidate in the population. These fitness values are the criteria on which selection algorithms operate that pick the most promising individuals for further investigation while discarding the lesser successful ones. The solution candidates which managed to enter the so-called *mating pool* are then reproduced, i. e., combined via crossover or slightly changed by mutation operations. After this is done, the cycle starts again in the next generation.

4.1 Search Space

When analyzing the problem structure outlined in Section 2, it becomes very obvious that standard encodings such as binary [23] or integer strings, matrixes, or real vectors cannot be used in the context of this special logistics planning task. Although it might be possible to create a genotype-phenotype mapping capable of translating an integer string into a tuple r representing a valid tour, trying to encode a set R of a variable number of such tours in an integer string is not feasible. First, there are many substructures of variable length such as the sets of orders $\underline{o}$ and swap bodies $\underline{b}$ involved in a tour. Also, it would practically be impossible to ensure the required *physical soundness* of the tours given that the reproduction operations would randomly modify the integer strings.

In our work, we adhered to the premise that *all solution candidates must represent correct, i. e., physically sound, solutions and none of the search operations is allowed to violate this correctness.* A solution candidate R not necessarily contains a complete plan which manages to deliver every order. Instead, partial solutions (as demanded in Section 1) are admitted, too. However, all individuals in the populations not only respect the laws of physics but also requirements such as cyclic routes for trucks.

In order to achieve such a behavior, it is clear that all reproduction operations of our evolutionary algorithm must have access to the complete tuples r. Therefore, the phenotypes are not encoded at all but instead, the plan objects in their native representations as illustrated in Figure 1.

4.2 Search Operations

By using this explicit representation, the search operations have full access to all information in the freight plans. Standard crossover and mutation operators are, however, no longer applicable. Instead, intelligent operators have to be introduced which respect the correctness of the solution candidates. In total, three crossover and sixteen mutation operations have been defined, each dealing with a specific constellation in the phenotypes and performing one distinct type of

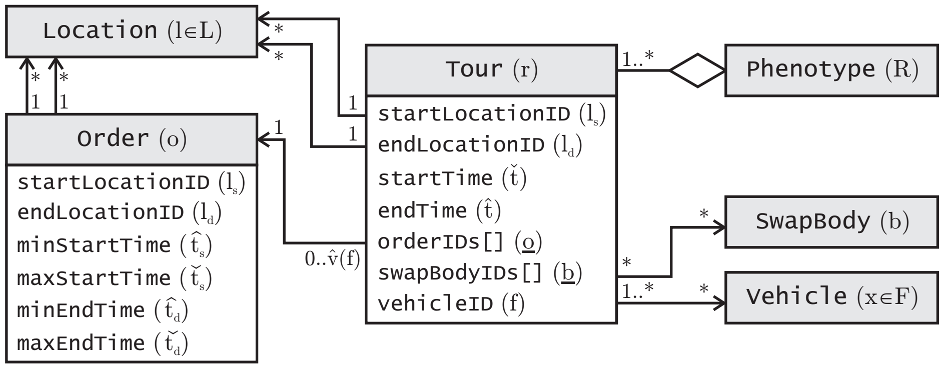

Fig. 1: The structure of the phenotypes R.

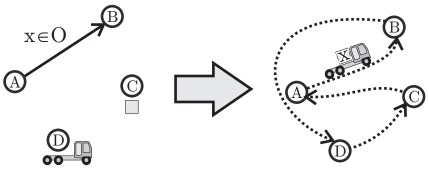

Fig. 2.1: Add an order.

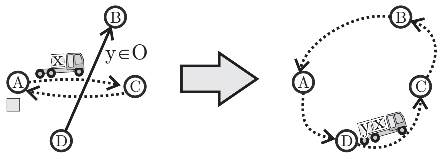

Fig. 2.2: Append an order.

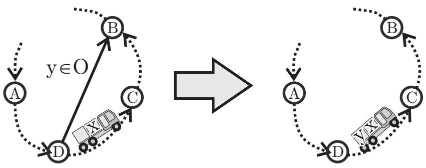

Fig. 2.3: Incorporate an order.

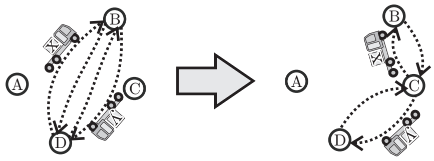

Fig. 2.4: Create a freight exchange.

Fig. 2: Some mutation operators from the freight planning EA.

modification. During the evolution, individuals to be mutated are processed by a randomly picked operator. If the operator is not applicable because the individual does not exhibit the corresponding constellation, another operator is tried and so on. Two individuals to be combined with crossover are processed by a randomly selected operator as well.

Obviously, we cannot give detailed specifications on all twenty genetic operations [24] (including the initial individual creation) in this paper. Instead, we will outline the four mutation operators sketched in Figure 2 exemplarily.

The first operator (Fig. 2.1) is applicable if there at least one order o which is not delivered if the plan in the input phenotype R was carried out. This operator chooses randomly from all available means of transportation. *Available* in this context means not involved in another tour for the time between the start and end time of the order. The freight transporters closer to the source of the order are picked with higher probability. Then, a swap body is allocated in the same manner. This process leads to between one and three new tours which are added to the phenotype. If the transportation vehicle is a truck, a fourth tour is added which travels back to the origin.

Fig. 2.2 illustrates one operator which extends a given set of tours by including a new order in the route of a vehicle. The mutator sketched in Fig. 2.3 does the same if an additional order can be included in already existing tours. For all operators which add new orders, swap bodies, or tours to the solution candidates, similar operations which remove these elements are provided.

Especially interesting is the “truck-meets-truck” mechanism. Often, two trucks are carrying out deliveries in opposite directions ($B \to D$ and $D \to B$ in Fig. 2.4). If the time windows of the orders allow it, the two involved trucks can meet at a halting point somewhere in the near of their routes and exchange their freight. This way, the total distance that they have to drive can almost be halved.

The crossover operators combine two phenotypes by intermixing their tours. In this process, tours which belong together such as those created in the first mutator are kept together.

4.3 Objective Functions

The freight transportation planning process run by the evolutionary algorithm is driven by three objective functions. These functions, all subject to minimization, are based on the requirements stated in Section 1 and are combined via Pareto comparisons [3, 21] in the subsequent fitness assignment processes.

- f_1: **Order Completion** One of the most important aspects of freight planning is to deliver as many of the orders as possible. Human operators would need to hire external carriers for orders which cannot be delivered (due to insufficient resources, for instance). Therefore, the first objective function returns the number of orders not considered in a freight plan.
- f_2: **Kilometers Driven** By using a distance matrix kept in memory, the second objective function determines the total distance covered by all vehicles involved. Minimizing this number will lead to less fuel consumption and thus, lower costs and lesser CO_2 production.
- f_3: **Full Utilization of the Capacities** The third objective function minimizes the spare capacities of the vehicles involved in tours. In other words, it counts the total volume left empty in the swap bodies on the road and the unused swap body slots of the trucks and trains.

5 Experiments

We have evaluated our freight planning system rigorously. Therefore, we have carried out a series of tests according to the full factorial design of experiments paradigm [25, 26]. These experiments (which we will discuss in Section 5.1) are based on a single real-world set of orders. The results of additional experiments performed with different datasets are outlined in Section 5.2. All data used has been reconstructed from the actual order database of the project partner *DHL*, one of the largest logistics companies worldwide.

The experiments were carried out using a simplified distance matrix for both, the EA and the original plans. Furthermore, legal aspects like statutory idle periods of the truck drivers have not been incorporated. However, an analysis of the results showed that these were not violated by the plans. The EA is able to utilize trains, but since the original plans did not do this, we turned off this feature in the experiments too, in order to keep the results comparable.

5.1 Full Factorial Tests

The full factorial test series is based on a set of 183 orders reconstructed from one day in December 2007. The original freight plan for these orders contained 159 tours which covered a total distance of $\mathbf{d} = 19\,109\,\text{km}$. The capacity of the vehicles involved was used to 65.5%.

ss The parent individuals in the population are either discarded (generational, $ss = 0$) or compete with their offspring (steady-state, $ss = 1$).
el The best solution candidates were either preserved (elitism, $el = 1$) or not (no elitism, $el = 0$).
ps Three different population sizes were tested: $ps \in \{200, 500, 1000\}$
fa Either simple Pareto-Ranking [3] ($fa = 0$) or an extended assignment process with sharing ($fa = 1$, called *variety preserving* in [21]) were applied.
cp The simple convergence prevention (SCP) method proposed in [21] was either used ($cp = 1$) or not ($cp = 0$).
mr/cr Different settings for the mutation rate $mr \in \{0.6, 0.8\}$ and the crossover rate $cr \in \{0.2, 0.4\}$ were tested.

These settings were varied in the experiments and each one of the 192 possible configurations was tested ten times. All runs utilized a tournament selection scheme with five parents and were granted 10 000 generations. The following measurements were collected:

ar The number of runs which found plans that completely covered all orders.
at The *median* number of generations needed by the runs succeeding in this to find such plans.
gr The number of runs which managed to find such plans which additionally were at least as good as the original freight plans.
gt The *median* number of generations needed by the runs succeeding in this to find such plans.
et The *median* number of generations after which f_2 did not improve by more than 1%, i. e., the point where the experiments could have been stopped without significant loss in the quality of the results.
eτ The *median* number of individual evaluations until this point.
d The *median* value of f_2, i. e., the median distance covered.

#	mr	cr	cp	el	ps	ss	fa	ar	at	gr	gt	et	eτ	d
1.	0.8	0.4	1	1	1000	1	1	10/10	341	10/10	609	3078	3 078 500	15 883 km
2.	0.6	0.2	1	0	1000	1	1	10/10	502	10/10	770	5746	5 746 500	15 908 km
3.	0.8	0.2	1	1	1000	1	1	10/10	360	10/10	626	4831	4 831 000	15 929 km
4.	0.6	0.4	1	0	1000	1	1	10/10	468	10/10	736	5934	5 934 000	15 970 km
5.	0.6	0.2	1	1	1000	1	1	10/10	429	10/10	713	6236	6 236 500	15 971 km
6.	0.8	0.2	1	0	1000	1	1	10/10	375	10/10	674	5466	5 466 000	16 003 km
7.	0.8	0.4	1	1	1000	1	0	10/10	370	10/10	610	5691	5 691 500	16 008 km
8.	0.8	0.2	1	0	1000	0	1	10/10	222	10/10	450	6186	6 186 500	16 018 km
9.	0.8	0.4	0	0	1000	0	1	10/10	220	10/10	463	4880	4 880 000	16 060 km
10.	0.8	0.2	0	1	1000	0	0	10/10	277	10/10	506	2862	2 862 500	16 071 km

Table 1: The top-ten evaluation results.

Table 1 contains the best ten configurations, sorted according to **gr**, **d**, and **e**τ. The best configuration managed to reduce the distance covered by over 3000 km (17%) on a constant basis. Even the configuration at rank 170 (not in Table 1) saved almost 1100 km in median. 172 out of the 192 test series managed to surpass the original plans for the orders in the data set in ten out of ten runs and only ten configurations were unable to achieve this goal at all.

We furthermore applied significance tests (sign and Wilcoxon's signed rank test [27, 21]) in order to test which parameter settings have significant positive influence. On a significance level of $\alpha = 0.02$, we considered a tendency only if both tests agreed. Applying the convergence prevention mechanism (SCP) [21], larger population sizes, variety preserving fitness assignment [21], elitism, and higher mutation and lower crossover rates have all significantly positive influence in general.

One run of the algorithm (prototypically implemented in Java) for this data set takes around three hours, for 1000 orders it still fulfills the requirement of delivering result in 24 hours. Runs with 3000 orders, however, take much longer. These measurements were taken on a single dual-core 2.6 GHz machine, which is only a fraction of the capacity available in the dedicated data centers of the project partners. It is well known that EAs can be efficiently distributed and thus, the final implementation will be able to deliver results in time for all situations.

5.2 Tests with Multiple Datasets

We have run experiments with many other real-world order datasets for which the actual freight plans used by the project partners were available. In all scenarios, our approach yielded an improvement which was never below 2.3%, usually above 7%, and for some days even reaching areas above 15%.

Figure 3 illustrates the best f_2-values (the total kilometers) of the individuals with the most orders satisfied in the population for two typical example

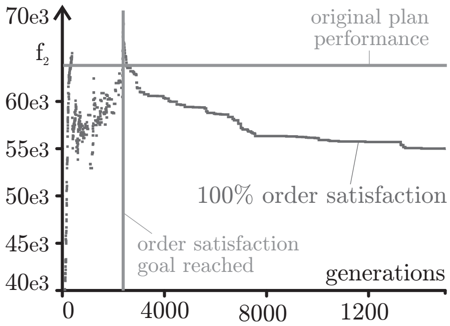

Fig. 3.1: For 642 orders (14% better).

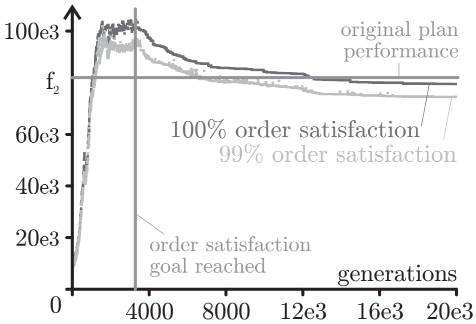

Fig. 3.2: For 1016 orders (3/10% better).

Fig. 3: Two examples for the freight plan evolution.

evolutions. In the two diagrams, the total distance first increases as the number of orders satisfied by the solution candidates increases. At some point, all orders are satisfied and now, the optimization of f_2 begins to kick in fully. Soon afterwards, the efficiency of the original plans is surpassed.

6 Conclusions

In this paper we presented the *INWEST* freight planning component which utilized an evolutionary algorithm with intelligent reproduction operations. Although this issue has not been elaborated on, the system returns a full Pareto frontier of the planning problem to the human operator, who can then make her choice according to the given circumstances. Our approach was tested rigorously on real-world data from the *INWEST* partners and achieved excellent results.

In the current phase, the component was implemented rather prototypically. A re-implementation in a more efficient manner will most likely lead to speed-up of a few percent. Additionally, features like online updates of the distance matrix which is used to compute f_2 and by the genetic operators for determining the time a truck needs to travel from one location to another, are planned. The system will then be capable to *a)* perform planning for the whole orders of one day in advance and *b)* update smaller portions of the plans online if traffic jams occur. Then, the system will be deployed in the computer centers of the project partners.

References

1. Weise, T., Podlich, A., Reinhard, K., Gorldt, C., Geihs, K.: Evolutionary freight transportation planning. In Giacobini, M., Machado, P., Brabazon, A., McCormack, J., Cagnoni, S., O'Neill, M., Caro, G.A.D., Neri, F., Ekárt, A., Preuss, M., Esparcia-Alcázar, A.I., Rothlauf, F., Farooq, M., Tarantino, E., Fink, A., Yang,

S., eds.: EvoTRANSLOG, 3rd European Workshop on Evolutionary Computation in Transportation and Logistics (Applications of Evolutionary Computing – Proceedings of EvoWorkshops 2009: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, EvoNUM, EvoSTOC, EvoTRANSLOG). Volume 5484/2009 of Lecture Notes in Computer Science (LNCS), Subseries: SL 1 – Theoretical Computer Science and General Issues., Berlin / Heidelberg, Springer (April 15–17, 2009) 768–777

2. Bundesministerium für Verkehr, Bau- und Stadtentwicklung: Verkehr in Zahlen 2006/2007. Deutscher Verkehrs-Verlag GmbH, Hamburg, Germany (2006)
3. Ceollo Coello, C.A., Lamont, G.B., van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Second edn. Volume 5 of Genetic and Evolutionary Computation. Kluwer Academic Publishers / Springer (1st ed: 2002, 2nd ed: 2007) Series Editor: David E. Goldberg and John R. Koza. Partly online available at http://books.google.de/books?id=sgX_Cst_yTsC [accessed 2008-10-20].
4. Technical Committee CEN/TC 119 "Swap body for combined transport": Swap bodies – non-stackable swap bodies of class C – dimensions and general requirements. EN 284:2006, CEN-CEN ELEC, Brussels, Belgium (November 10, 2006) See also: DIN 70013/14, BS 3951:Part 1, ISO 668, ISO 1161, ISO 6346, 88/218/EEC, UIC 592-4, UIC 596-6.
5. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & Operations Research **13**(5) (1986) 533–549 Online available at http://dx.doi.org/10.1016/0305-0548(86)90048-1 [accessed 2008-03-27].
6. Amberg, A., Domschke, W., Voß, S.: Multiple center capacitated arc routing problems: A tabu search algorithm using capacitated trees. European Journal of Operational Research (EJOR) **124**(2) (July 16, 2000) 360–376 Online available at http://dx.doi.org/10.1016/S0377-2217(99)00170-8 [accessed 2008-10-27].
7. Badeau, P., Gendreau, M., Guertin, F., Potvin, J.Y., Taillard, É.D.: A parallel tabu search heuristic for the vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies **5**(2) (April 1997) 109–122
8. Bräysy, O., Gendreau, M.: Tabu search heuristics for the vehicle routing problem with time windows. TOP: An Official Journal of the Spanish Society of Statistics and Operations Research **10**(2) (December 2002) 211–237 Online available at http://www.springerlink.com/content/854188435g26v0h5/fulltext.pdf [accessed 2008-10-27].
9. Breedam, A.V.: An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints. PhD thesis, University of Antwerp, Antwerp, Belgium (1994) en analyse van het gedrag van rittenplanningsheuristieken voor een selektie van planningsproblemen met vrachtwagengebonden, klantgebonden en tijdgebonden randvoorwaarden.
10. Czech, Z.J., Czarnas, P.: Parallel simulated annealing for the vehicle routing problem with time windows. In: 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (PDP'02), IEEE Computer Society (January 9–11, 2002) 376–383 Online available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.5766 [accessed http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.100.8805]2008-10-27.
11. Bullnheimer, B., Hartl, R.F., Strauss, C.: An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research **89**(0) (January 1999) 319–328 Online available at http://citeseerx.ist.

psu.edu/viewdoc/summary?doi=10.1.1.49.1415 and http://www.macs.hw.ac.uk/~dwcorne/Teaching/bullnheimer-vr.pdf [accessed 2008-10-27].
12. Doerner, K., Gronalt, M., Hartl, R.F., Reimann, M., Strauss, C., Stummer, M.: Savings ants for the vehicle routing problem. In: Proceedings of the Applications of Evolutionary Computing on EvoWorkshops 2002. (2002) 11–20 In proceedings [28]. Also: Report Series SFB "Adaptive Information Systems and Modelling in Economics and Management Science", Nr. 63, December 2001, Vienna University of Economics and Business Administration, Augasse 2–6, 1090 Wien, Austria, Dokument ID: oai:epub.wu-wien.ac.at:epub-wu-01_1f1, online available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.8882 and http://epub.wu-wien.ac.at/dyn/virlib/wp/mediate/epub-wu-01_1f1.pdf?ID=epub-wu-01_1f1 [accessed 2008-10-27].
13. rong Jih, W., jen Hsu, J.Y.: Dynamic vehicle routing using hybrid genetic algorithms. In: IEEE International Conference on Robotics and Automation. Volume 1. (May 10–15, 1999) 453–458 Online available at http://neo.lcc.uma.es/radi-aeb/WebVRP/data/articles/DVRP.pdf [accessed 2008-10-27].
14. Thangiah, S.R.: Vehicle routing with time windows using genetic algorithms. In: Practical Handbook of Genetic Algorithms: New Frontiers. CRC Press, Inc. (1995) 253–277 In collection [29]. Online available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.7903 [accessed 2008-10-27].
15. Zhu, K.Q.: A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. In: 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03), Washington, DC, USA, IEEE Computer Society (November 3–5, 2003) 176–183
16. Alba, E., Dorronsoro, B.: Solving the vehicle routing problem by using cellular genetic algorithms. In: Proceedings of the 4th European Conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP 2004. (2004) 11–20 In proceedings [30]. Online available at http://www.lcc.uma.es/~eat/pdf/evocop04.pdf [accessed 2008-10-27].
17. Ralphs, T.: Vehicle routing data sets (October 3, 2003) Data sets online available at http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/ [accessed 2008-10-27].
18. Pankratz, G., Krypczyk, V.: Benchmark data sets for dynamic vehicle routing problems (June 12, 2007) Data sets online available at http://www.fernuni-hagen.de/WINF/inhfrm/benchmark_data.htm [accessed 2008-10-27].
19. Weise, T., Zapf, M., Chiong, R., Nebro Urbaneja, A.J.: Why is optimization difficult? In: Nature-Inspired Algorithms for Optimisation. Springer (2009) In collection [31].
20. Radcliffe, N.J.: The algebra of genetic algorithms. Annals of Mathematics and Artificial Intelligence **10**(4) (December 1994) 339–384 Online available at http://www.springerlink.com/content/w20613t556788tr0/fulltext.pdf [accessed 2008-08-10]. Also technical report TR92-11, 1992 from Edinburgh Parallel Computing Centre, University of Edinburgh, King's Buildings, EH9 3JZ, Scotland. Online available at http://citeseer.ist.psu.edu/radcliffe94algebra.html and http://users.breathe.com/njr/formaPapers.html [accessed 2007-07-28].
21. Weise, T.: Global Optimization Algorithms – Theory and Application. second edn. Self-Published (2017) Online available at http://www.it-weise.de/ [accessed 2017-03-23].
22. Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press US (January 1996) Partly online available at http://books.google.de/books?id=EaN7kvl5coYC [accessed 2009-02-03].

23. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA (January 1989)
24. Podlich, A.: Intelligente planung und optimierung des güterverkehrs auf straße und schiene mit evolutionären algorithmen. Master's thesis, University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany (March 2009) Advisor: Thomas Weise, Christian Gorldt, Manfred Menze. Supervisor: Prof. Dr. Kurt Geihs, Prof. Dr.-Ing. B. Scholz-Reiter.
25. Box, G.E.P., Hunter, J.S., Hunter, W.G.: Statistics for Experimenters: Design, Innovation, and Discovery. John Wiley & Sons (May 1st ed: 1978, 2nd ed: 2005)
26. Yates, F.: The Design and Analysis of Factorial Experiments. Imperial Bureau of Soil Science, Commonwealth Agricultural Bureaux, Harpenden, England, UK (1937) Technical Communication No. 35.
27. Siegel, S., Castellan Jr., N.J.: Nonparametric Statistics for The Behavioral Sciences. Humanities/Social Sciences/Languages. McGraw-Hill Book Company, New York, NY, USA (1st: 1956, 2nd: 1988)
28. Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R., eds.: Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2002: EvoCOP, EvoIASP, EvoSTIM/EvoPLAN. In Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R., eds.: EvoWorkshops. Volume 2279/-1/2002 of Lecture Notes in Computer Science (LNCS)., Springer (April 3–4, 2002)
29. Chambers, L.D., ed.: Practical Handbook of Genetic Algorithms: New Frontiers. Volume II of Practical Handbook of Genetic Algorithms. CRC Press, Inc., Boca Raton, FL, USA (1995) Partly online available at `http://books.google.de/books?id=9RCE3pgj9K4C` [accessed 2008-07-19].
30. Gottlieb, J., Raidl, G.R., eds.: Proceedings of the 4th European Conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP 2004. In Gottlieb, J., Raidl, G.R., eds.: EvoCOP. Volume 3004/2004 of Lecture Notes in Computer Science (LNCS)., Springer (April 5–7, 2004)
31. Chiong, R., ed.: Nature-Inspired Algorithms for Optimisation. Volume 193 of Studies in Computational Intelligence. Springer (April 2009)

```
@inproceedings{WPRGG2009EFTP,
  author        = {Thomas Weise and Alexander Podlich and Kai Reinhard
                   and Christian Gorldt and Kurt Geihs},
  title         = {Evolutionary Freight Transportation Planning},
  booktitle     = {EvoTRANSLOG, 3rd European Workshop on Evolutionary
                   Computation in Transportation and Logistics
                   (Applications of Evolutionary Computing -- Proceedings
                   of EvoWorkshops 2009: EvoCOMNET, EvoENVIRONMENT, EvoFIN,
                   EvoGAMES, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART,
                   EvoNUM, EvoSTOC, EvoTRANSLOG)},
  month         = apr # {~15--17, },
  year          = {2009},
  location      = {T{\"u}bingen, Germany},
  doi           = {10.1007/978-3-642-01129-0_87},
  pages         = {768--777},
  series        = {Lecture Notes in Computer Science (LNCS), Subseries:
                   SL 1 -- Theoretical Computer Science and General Issues},
  volume        = {5484/2009},
  publisher     = {Springer},
  address       = {Berlin / Heidelberg},
  ISSN          = {0302-9743 (Print) 1611-3349 (Online)},
  editor        = {Mario Giacobini and Penousal Machado and Anthony Brabazon
                   and Jon McCormack and Stefano Cagnoni and Michael O'Neill
                   and Gianni A. Di Caro and Ferrante Neri and
                   Anik{\'o} Ek{\'a}rt and Mike Preuss and
                   Anna I Esparcia-Alc{\'a}zar and Franz Rothlauf and
                   Muddassar Farooq and Ernesto Tarantino and Andreas Fink
                   and Shengxiang Yang},
  isbn          = {3-642-01128-4, 978-3-642-01128-3},
}
```