Solving Real-World Vehicle Routing Problems with Evolutionary Algorithms

Thomas Weise and Alexander Podlich and Christian Gorldt

Abstract In this chapter, we present the freight transportation planning component of the *in.west* project. This system uses 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.

Thomas Weise
Distributed Systems Group, University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany, e-mail: weise@vs@uni-kassel.de

Alexander Podlich
Micromata GmbH Kassel, Marie-Calm-Straße 3, 34131 Kassel, Germany, e-mail: a.podlich@micromata.de

Christian Gorld
BIBA – Bremer Institut für Produktion und Logistik GmbH, Hochschulring 20, 28359 Bremen, Germany e-mail: gor@biba.uni-bremen.de

1 Introduction

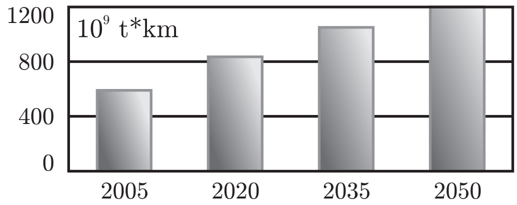

Fig. 1: The freight traffic on German roads in billion tons*kilometer.

According to the German Federal Ministry of Economics and Technology [14], the freight traffic volume on German roads will have doubled by 2050 as illustrated in Figure 1. Reasons for this development are the effects of globalization as well as the central location of the country in Europe. With the steadily increasing freight traffic resulting from trade inside the European Union and global import and export [13], transportation and logistics become more important [7, 45]. Thus, a need for intelligent solutions for the strategic planning of logistics becomes apparent [14]. Such a planning process can be considered as a multi-objective optimization problem which has the goals [49, 54] of increasing the profit of the logistics companies by

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

Fortunately, 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 via (inexpensive) trains, not only the driver's work hours and the costs are decreased, but also the CO_2 production declines.

Efficient freight planning is not a static procedure. Although it involves building an overall plan on how to deliver orders, it should also be able to dynamically react to unforeseen problems such as traffic jams or accidents. This reaction should lead to a local adaptation of the plan and re-routing of all involved freight vehicles whereas parts of the plan concerning geographically distant and uninvolved objects are supposed to stay unchanged.

In the literature, the creation of freight plans is known as the *Vehicle Routing Problem.* In this chapter, we present an approach to Vehicle Routing for real-world scenarios: the freight transportation planning component of the *in.west* system. *in.west*, or "Intelligente Wechselbrücksteuerung" in full, is a joint research project of *DHL*, *Deutsche Post AG*, *Micromata*, *BIBA*, and *OHB Teledata* funded by the German Federal Ministry of Economics and Technology.[1]

In the following section, we discuss different flavors of the Vehicle Routing Problem and the general requirements of logistics departments which specify the framework for our freight planning component. These specific conditions rendered the related approaches outlined in Section 3 infeasible for our situation. In Section 4, we present an Evolutionary Algorithm for multi-objective, real-world freight planning problems [40]. 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. Our approach has been tested in many different scenarios and the experimental results are summarized in Section 5. The freight transportation planning component described in this chapter is only one part of the holistic *in.west* approach to logistics which will be outlined in Section 6. Finally, we conclude with a discussion of the results and future work in Section 7.

2 Vehicle Routing in Theory and Practice

2.1 Vehicle Routing Problems

The Vehicle Routing Problem (VRP) is one of the most famous combinatorial optimization problems. In simple terms, the goal is to determine a set of routes than can satisfy several geographically scattered customers' demands while minimizing the overall costs [37]. Usually, a fleet of vehicles located in one depot is supposed to fulfill these requests. In this context, the original version of the VRP problem was proposed by Dantzig and Ramser [21] in 1959 who addressed the calculation of a set of optimal routes for a fleet of gasoline delivery trucks.

As described next, a large number of variants of the VRP exist, adding different constraints to the original definition. Within the scope of *in.west*, we first identified all the restrictions of real-world Vehicle Routing Problems that occur in companies like *DHL* and then analyzed available approaches from the literature.

The *Capacitated Vehicle Routing Problem* (CVRP), for example, is similar to the classical Vehicle Routing Problem with the additional constraint that every vehicle must have the same capacity. A fixed fleet of delivery vehicles must service known customers' demands of a single commodity from a common depot at minimum transit costs [25, 44, 41]. The *Distance Vehicle Routing Problem* (DVRP) is a VRP extended with the additional constraint on the maximum total distance traveled by each vehicle. In addition, *Multiple*

[1] See `http://www.inwest.org/` [accessed 2008-10-29].

Depot Vehicle Routing Problems (MDVRP) have several depots from which customers can be supplied. Therefore, the MDVPR requires the assignment of customers to depots. A fleet of vehicles is based at each depot. Each vehicle then starts at its corresponding depot, services the customers assigned to that depot, and returns.

Typically, the planning period for a classical VRP is a single day. Different from this approach are *Periodic Vehicle Routing Problems* (PVRP), where the planning period is extended to a specific number of days and customers have to be served several times with commodities. In practice, *Vehicle Routing Problems with Backhauls* (VRPB), where customers can return some commodities [41] are very common. Therefore all deliveries for each route must be completed before any pickups are made. Then, it also becomes necessary to take into account that the goods which customers return to the deliverer must fit into the vehicle.

The *Vehicle Routing Problem with Pick-up and Delivering* (VRPPD) is a capacitated Vehicle Routing Problem where each customer can be supplied with commodities as well as return commodities to the deliverer. Finally the *Vehicle Routing Problem with Time Windows* (VRPTW) is similar to the classical Vehicle Routing Problem with the additional restriction that time windows (intervals) are defined in which the customers have to be supplied [41]. Figure 2 shows the hierarchy of Vehicle Routing Problem variants and also the problems which are relevant in the *in.west* case.

2.2 Model of a Real-World Situation

As it becomes obvious from Figure 2, the situation in logistics companies is relatively complicated and involves many different aspects of Vehicle Routing. The basic unit of freight considered in this work is a swap body b, a standardized container (C 745, EN 284 [15]) with a dimension 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 *DHL* usually own up to one thousand such containers. We refer to the union of all swap bodies as the set B.

We furthermore define the union of all possible means of transportation as the set F. All trucks $tr \in F$ can carry at most a certain maximum number $\hat{v}(tr)$ of swap bodies at once. Commonly and also in the case of *DHL*, this limit is $\hat{v}(tr) = 2$. The maximum load of trains $z \in F$, on the other hand, is often more variable and usually ranges somewhere between 30 and 60 ($\hat{v}(z) \in [30..60]$). Trains have fixed routes, departure, and arrival times whereas freight trucks can move freely on the map. In many companies, trucks must perform cyclic tours, i.e., return to their point of departure by the end of the day, in order to allow the drivers to return home.

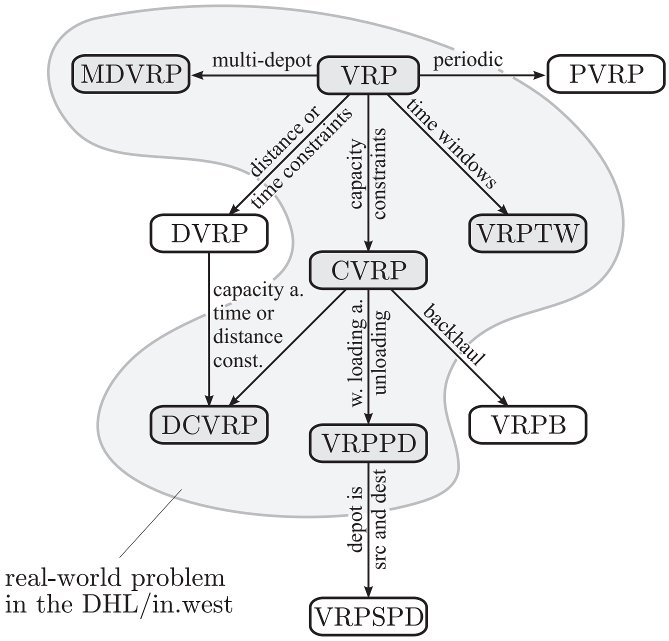

Fig. 2: Different flavors of the VRP and their relation to the *in.west* system.

The clients and the depots of the logistics companies together can form more than one thousand locations from which freight may be collected 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 $[\breve{t_s}, \widehat{t_s}]$ in which it must be collected from its source $l_s \in L$. From there, it has to be carried to its destination location $l_d \in L$ where it must arrive within a time window $[\breve{t_d}, \widehat{t_d}]$. An order furthermore has a volume v which we assume to be an integer multiple of the capacity of a swap body. Hence, a transportation order o can fully be described by the tuple $o = \left\langle l_s, l_d, [\breve{t_s}, \widehat{t_s}], [\breve{t_d}, \widehat{t_d}], v \right\rangle$. In our approach, orders which require more than one ($v > 1$) swap body will be split up into multiple orders requiring one swap body ($v = 1$) each.

Logistics companies usually have to service up to a few thousand such orders per day. The *express* unit of the project partner *DHL*, for instance, delivered between 100 and 3000 per day in 2007, depending on the day of the week as well as national holidays etc.

The result of the planning process is a *set* X of tours. Each single tour x is described by a tuple $x = \langle l_s, l_d, f, \check{t}, \hat{t}, \underline{b}, \underline{o} \rangle$. l_s and l_d are the start and destination locations and $\check{t}$ and $\hat{t}$ are the departure and arrival time of the vehicle $f \in F$. On this tour, f carries the set $\underline{b} = \{b_1, b_2, \dots\}$ of swap bodies which, in turn, contain the orders $\underline{o} = \{o_1, o_2, \dots\}$. It is assumed that, for each truck, there is at least one corresponding truck driver and that the same holds for all trains.

Tours are the smallest unit of freight transportation. Usually, multiple tours are combined for a delivery: First, a truck tr may need to drive from the depot in Dortmund to Bochum to pick up an unused swap body sb ($x_1 = \langle \text{Dortmund}, \text{Bochum}, tr, \text{9am}, \text{10am}, \emptyset, \emptyset \rangle$). In a subsequent tour $x_2 = \langle \text{Bochum}, \text{Essen}, tr, \text{10.05am}, \text{11am}, \{sb\}, \emptyset \rangle$, it carries the empty swap body sb to a customer in Essen. There, the order o is loaded into sb and then transported to its destination $o.l_d$ = Hannover ($x_3 = \langle \text{Essen}, \text{Hannover}, tr, \text{11.30am}, \text{4pm}, \{sb\}, \{o\} \rangle$).

Obviously, the set X must be physically sound. It must, for instance, not contain any two intersecting tours x_1, x_2 $\left(\left(x_1.\check{t} < x_2.\hat{t}\right) \wedge \left(x_2.\check{t} < x_1.\hat{t}\right)\right)$ involving the same vehicle ($x_1.f = x_2.f$), swap bodies ($x_1.\underline{b} \cap x_2.\underline{b} \neq \emptyset$), or orders ($x_1.\underline{o} \cap x_2.\underline{o} \neq \emptyset$). Also, it must be ensured that all objects involved in a tour x reside at $x.l_s$ at time $x.\check{t}$. Furthermore, the capacity limits of all involved means of transportation must be respected, i.e., $0 \leq |x.\underline{b}| \leq \hat{v}(x.f)$. 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. The same goes for laws restricting the maximum amount of time a truck driver is allowed to drive without breaks and constraints imposed by the company's policies such as the aforementioned cyclic character of truck tours. Only plans for which all these conditions hold can be considered as *correct*.

From the perspective of the planning system's user, runtime constraints are of the same importance: Ideally, the optimization process should not exceed one day. Even the best results become useless if their computation takes longer than the time span from receiving the orders to the day where they actually have to be delivered.

Experience has shown that hiring external carriers for a small fraction of the freight can often reduce the number of required tours to be carried out by the own vehicles and the corresponding total distance to be covered significantly, if the organization's existing capacities are already utilized to their limits. Therefore, a good transportation planning system should also be able to make suggestions on opportunities for such an on-demand outsourcing. An example for this issue is illustrated in Fig. 6.2.

The framework introduced in this section holds for practical scenarios in logistics companies like *DHL* and Deutsche Post. It proposes a hard challenge for research, since it involves multiple intertwined optimization problems and combines several aspects even surpassing the complexity of the most difficult Vehicle Routing Problems known from the literature.

3 Related Work

The approaches discussed in the literature on freight transportation planning can roughly be divided into two basic families: exact and stochastic or metaheuristic methods. The exact approaches are usually only able to solve small instances of Vehicle Routing Problems – i. e., those with very limited numbers of orders, customers, or locations – and therefore cannot be applied in most real-world situations. Using them in scenarios with many constraints further complicates the problem. [37]

Heuristic methods are reliable and efficient approaches to address Vehicle Routing Problems of larger scale. Despite the growing problem dimension, they are still able to provide high quality approximate solutions in a reasonable time. This makes them more attractive than exact methods for practical applications. In over 40 years of research, a large number of heuristics have been proposed for VRPs.

Especially, the metaheuristic optimization methods have received more and more attention. Well-known members of this family of algorithms which have been applied to Vehicle Routing and freight transportation planning are Tabu Search [26, 3, 6, 10], Simulated Annealing [11, 20], Ant Systems [12, 24], and particularly Evolutionary Algorithms [31, 51, 58, 1, 9].

Ombuki-Berman and Hanshar [35], for example, proposed a Genetic Algorithm (GA) for a Multiple Depot Vehicle Routing Problem. They therefore adopted an indirect and adaptive inter-depot mutation exchange strategy, coupled with capacity and route-length restrictions.

Machado et al. [33] used a basic Vehicle Routing Problem to compare a standard evolutionary approach with a coevolutionary method. They showed that the inclusion of a heuristic method into evolutionary techniques significantly improves the results. Instead of using additional heuristics, knowledge of the problem domain is incorporated into the search operations in our work.

A cellular and thus, decentralized, GA for solving the Capacitated Vehicle Routing Problem was presented by Alba and Dorronsoro [1, 2]. This method has a high performance in terms of the quality of the solutions found and the number of function evaluations needed. Decentralization is a good basis for distributing EAs, a method for speeding up the evolution which we will consider in our future work.

These methods perform a single-objective optimization enriched with problem-specific constraints. The size of the problems tackled is roughly around a few hundred customers and below 1000 orders. This is the case in most of the test sets available. Examples of such benchmarks are the datasets by Augerat et al. [4], Van Breedam [11], Golden et al. [28], Christofides et al. [18], and Taillard [50] which are publicly available at [43, 23, 36]. Using these (partly artificial) benchmarks in our work was not possible since the framework conditions in *in.west* are very different. Therefore, we could not perform a direct comparison of our system with the other approaches mentioned.

To our knowledge, the problem most similar to the practical situation specified in Section 2.2 is the *Multiple Depot Vehicle Routing Problem with Pickup, Delivery and Intermediary Depots* (MDVRPPDID) defined by Sigurjónsson [48]. This problem, however, does not consider orders and freight containers as different objects. Instead, each container has a source and a target destination and corresponds to one order. Also, all vehicles have the same capacity of one container which is not the case in our system where trucks can usually transport two containers and trains have much higher capacities. The Tabu Search approach developed by Sigurjónsson [48] is similar to our method in that it incorporates domain knowledge in the solution structure and search operations. However, it also allows infeasible intermediate solutions which we rule out in Section 4. It was tested on datasets with up to 16 depots, 40 vehicles, and 100 containers which is more than a magnitude smaller than the problem dimensions the *in.west* system has to deal with.

Confessore et al. [19] define a Genetic Algorithm for the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW, see Figure 2) for real-world scenarios with a heterogeneous vehicle fleet with different capacities, multi-dimensional capacity constraints, order/vehicle, item/vehicle, and item/item compatibility constraints. In *in.west*, the heterogeneity of the vehicles is taken a step further in the sense that trains have totally different characteristics in terms of the degrees of freedom regarding the tour times and end points. Furthermore, in *in.west*, orders are not assigned to vehicles but to containers which, in turn, are assigned to trucks and trains.

The general idea of using Evolutionary Algorithms and their hybrids for Vehicle Routing Problems has proven to be very efficient [41]. The quality of solutions produced by evolutionary or genetic methods is often higher than that obtained by classic heuristics. Potvin [41] pointed out that Evolutionary Algorithms can also outperform widely used metaheuristcs like Tabu Search on classic problems. He also states that other approaches like Artificial Neural Networks have more or less been abandoned by now in the area of VRPs due to their poor performance on off-the-shelf computer platforms.

In many of the publications listed in this section, it is indicated that metaheuristics work best when a good share of domain knowledge is incorporated. This holds not only for Vehicle Routing, but also in virtually every other application of global optimization [52, 55, 42]. 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 at the center of the approach.

4 Evolutionary Approach

4.1 Evolutionary Algorithms

Evolutionary Algorithms (EAs) 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 $X \in \mathbb{X}$ from the search space $\mathbb{X}$ iteratively [52, 5]. Their goal is to find the element(s) $X^{\star} \in \mathbb{X}$ for which the objective function $f : \mathbb{X} \mapsto \mathbb{R}$ takes on the optimal values. Evolutionary Algorithms which work on multiple such functions $F = \{f_1, f_2, .., f_n\}$ are called multi-objective Evolutionary Algorithms (MOEAs) [16, 17].

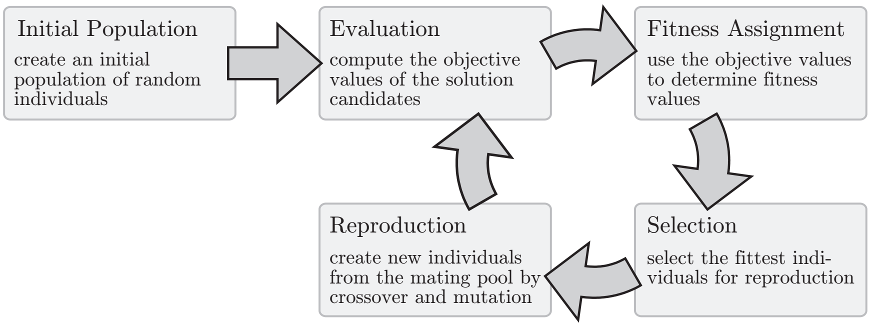

Fig. 3: The basic cycle of Evolutionary Algorithms.

All EAs proceed according to the schema depicted in Figure 3. First, an initial population of randomly configured individuals is created. Every iteration then starts with the evaluation of the objective functions on the individuals in the population. Based on their results, a relative fitness is assigned to each solution candidate in the population. These fitness values are the criteria on which selection algorithms operate to pick the most promising individuals for further investigation while discarding the less 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.

4.2 Search Space

When analyzing the problem structure outlined in Section 2.2, it becomes very obvious that standard encodings such as binary [27] or integer strings, matrixes, or real vectors cannot be used in the context of this very general logistics planning task. Although it might be possible to create a genotype-phenotype mapping capable of translating an integer string into a tuple x representing a valid tour, trying to encode a set X of a variable number of such tours in an integer string is not feasible. First, there are many sub-structures involved in a tour which have variable length such as the sets of orders $\underline{o}$ and swap bodies $\underline{b}$. Second, it would be practically 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 solutions* according to the specification given in Section 2.2 and *none of the search operations are allowed to violate this correctness*. A solution candidate $X \in \mathbb{X}$ does not necessarily contain a complete plan which manages to deliver all orders. Instead, partial solutions (again as demanded in Section 2.2) are admitted, too.

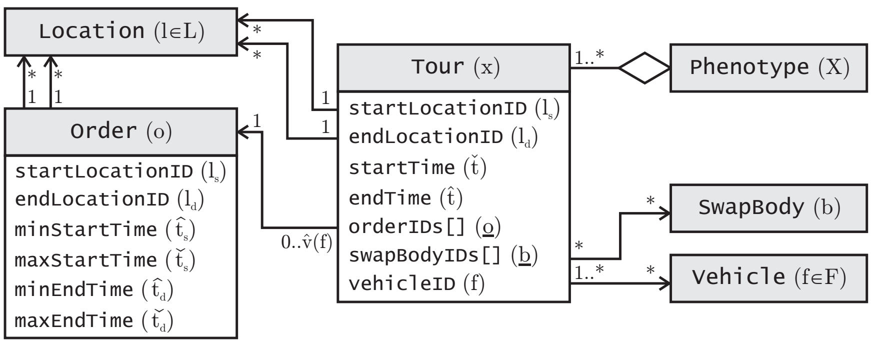

Fig. 4: The structure of the phenotypes X.

In order to achieve such a behavior, it is clear that all reproduction operations in the Evolutionary Algorithm must have access to the complete set X of tuples x. Only then, they can check whether the modifications to be applied may impair the correctness of the plans. Therefore, the phenotypes are not encoded at all, but instead, they are the plan objects in their native representations as illustrated in Figure 4.

This figure holds the UML specification of the phenotypes in our planning system. The exactly same data structures are also used by the *in.west* middleware and graphical user interface. The location IDs (`startLocationID`, `endLocationID`) of the `Orders` and `Tours` are indices into a database. They are also used to obtain distances and times of travel between locations from a sparse distance matrix which can be updated asynchronously from different information sources. The `orderIDs`, `swapBodyIDs`, and `vehicleIDs` are indices into a database as well.

4.3 Search Operations

By using this explicit representation, the search operations have full access to all the 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.

For the *in.west* planning system, three crossover and sixteen mutation operations have been defined, each dealing with a specific constellation in the phenotypes and performing one distinct type of 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 belong to the corresponding constellation, another operator is tried. This is repeated, until either the individual is modified or all operators were tested. Two individuals to be combined with crossover are processed by a randomly selected operator as well.

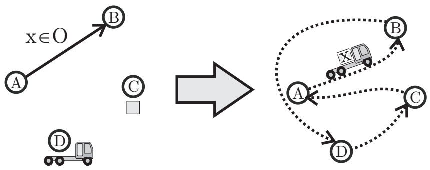

Fig. 5.1: Add an order.

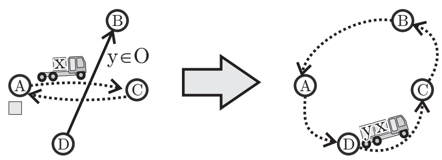

Fig. 5.2: Append an order.

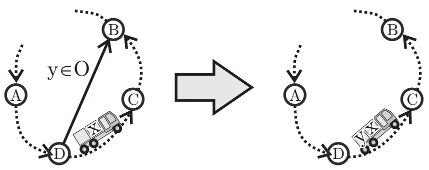

Fig. 5.3: Incorporate an order.

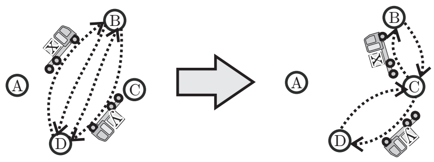

Fig. 5.4: Create a freight exchange.

Fig. 5: Some mutation operators from the freight planning EA.

Obviously, we cannot give detailed specifications on all twenty genetic operations [39] (including the initial individual creation) in this chapter. Instead, we will outline the mutation operators sketched in Figure 5 exemplarily.

The first operator (Fig. 5.1) is applicable if there is at least one order which would not be delivered if the plan in the input phenotype X 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 times 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 being added to the phenotype. If the transportation vehicle is a truck, a fourth tour is added which allows it to travel back to its starting point. This step is optional and is applied only if the system is configured to send all trucks back "home" after the end of their routes, as it is the case in *DHL*.

Fig. 5.2 illustrates one operator which tries to include an additional order o into an already existing set of related tours. If the truck driving these tours has space for another swap body, at least one free swap body b is available, and picking up o and b as well as delivering o is possible without violating the time constraints of the other transportation orders already involved in the set of tours, the order is included and the corresponding new tours are added to the plan.

The mutator sketched in Fig. 5.3 does the same if an additional order can be included in already existing tours because of available capacities in swap bodies. Such spare capacities occur from time to time since the containers are not left at the customers' locations after unloading the goods but transported back to the depots. For all operators which add new orders, swap bodies, or tours to the solution candidates, inverse operations which remove these elements are provided, too.

One exceptional operator 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. 5.4). The operator tries to find a location C which is close to both, B and D. If the time windows of the orders allow it, the two involved trucks can meet at this halting point C and exchange their freight. This way, the total distance that they have to drive can almost be halved from $4 * \overline{BD}$ to $2 * \overline{BC} + 2 * \overline{CD}$ where $\overline{BC} + \overline{CD} \approx \overline{BD}$.

The first recombination operator used in the *in.west* system copies all tours from the first parent and then adds all tours from the second parent in a way that does not lead to a violation of the solution's correctness. In this process, tours which belong together such as those created by the first mutator mentioned are kept together. A second crossover method tries to find sets of tours in the first parent which intersect with similar sets in the second parent and joins them into an offspring plan in the same way the truck-meets-truck mutator combines tours.

4.4 Objective Functions

The freight transportation planning process run by the Evolutionary Algorithm is driven by a set F of three objective functions ($F = \{f_1, f_2, f_3\}$). These functions, all subject to minimization, are based on the requirements stated in Section 1 and are combined via Pareto comparisons [52, 16, 17] in the fitness assignment processes.

4.4.1 f_1: Order Delivery

One of the most important aspects of freight planning is to deliver as many orders as possible. Therefore, the first objective function $f_1(X)$ returns the number of orders which will not be delivered in a timely manner if the plan X was carried out. The optimum of f_1 is zero. Human operators need to hire external carriers for orders which cannot be delivered (due to insufficient resources, for instance).

4.4.2 f_2: Kilometers Driven

By using a sparse distance matrix stored in memory, the second objective function determines the total distance covered by all vehicles involved. Minimizing this distance will lead to less fuel consumption and thus, lower costs and lesser CO_2 production. The global optimum of this function is not known a priori and may not be discovered by the optimization process as well.

4.4.3 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 considers the total volume left empty in the swap bodies on the road and the unused swap body slots of the trucks and trains. f_2 does not consider whether trucks are driving tours empty or loaded with empty containers. These aspects are handled by f_3 which again has the optimum zero.

5 Experiments

Because of the special requirements of the *in.west* project and the many constraints imposed on the corresponding optimization problem, the experimental results cannot be directly compared with other works. As we have shown in our discussion of related work in Section 3, none of the approaches in the Vehicle Routing literature are sufficiently similar to this scenario.

Hence, it was especially important to evaluate our freight planning system rigorously. We have therefore carried out a series of tests according to the full factorial design of experiments paradigm [8, 57]. 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 have been reconstructed from the actual order database of the project partner *DHL*, one of the largest logistics companies worldwide. This database is also the yardstick with which we have measured the utility of our system.

The experiments were conducted using a simplified distance matrix for both, the EA and the original plans. Since the original plans did not involve trains, we deactivated the mutation operators which incorporate train tours into solution candidates too – otherwise the results would have been incomparable. Legal aspects like statutory idle periods of the truck drivers have not been considered in the reproduction operators either. However, only plans not violating these constraints were considered in the experimental evaluation.

5.1 Full Factorial Tests

Evolutionary Algorithms have a wide variety of parameters, ranging from the choice of sub-algorithms (like those computing a fitness value from the vectors of objective values for each individual) to the mutation rate determining the fraction of the selected solution candidates which are to undergo mutation. The performance of an EA strongly depends on the configuration of these parameters. In different optimization problems, usually different configurations are beneficial and a setting finding optimal solutions in one application may lead to premature convergence to a local optimum in other scenarios. Because of the novelty of the presented approach for transportation planning, performing a large number of experiments with different settings of the EA was necessary in order to find the optimal configuration to be utilized in the *in.west* system in practice.

We, therefore, decided to conduct a full factorial experimental series, i. e., one where all possible combinations of settings of a set of configuration parameters are tested. As basis for this series, we used a test case consisting of 183 orders reconstructed from one day in December 2007. The original freight plan X_o for these orders contained 159 tours which covered a total distance of $d = f_2(X_o) = 19\,109\,\mathrm{km}$. The capacity of the vehicles involved was filled to 65.5%. The parameters examined in these experiments are listed in Table 1. These settings were varied in the experiments and each of the 192 possible configurations was tested ten times. All runs utilized a tournament selection scheme with five contestants and were granted 10 000 generations. The measurements collected are listed in Table 2.

Table 3 contains the thirteen best and the twelve worst configurations, sorted according to $\mathbf{gr}$, $\mathbf{d}$, and $\mathbf{e}\boldsymbol{\tau}$. The best configuration managed to reduce the distance to be covered by over 3000 km (17%) consistently. Even the configuration ranked 170 (not in Table 3) saved almost 1100 km in median. In total, 172 out of the 192 test series managed to surpass the original plans for the orders in the dataset in all of ten runs and only ten configurations were unable to achieve this goal at all.

Param.	Setting and Meaning
ss	In every generation of the EA, new individuals are created by the reproduction operations. The parent individuals in the population are then either discarded (generational, $ss = 0$) or compete with their offspring (steady-state, $ss = 1$).
el	Elitist Evolutionary Algorithms keep an additional archive preserving the best solution candidates found ($el = 1$). Using elitism ensures that these solution candidates cannot be lost due to the randomness of the selection process. Turning off this feature ($el = 0$) may allow the EA to escape local optima easier.
ps	Allowing the EA to work with populations consisting of many individuals increases its chance of finding good solutions but also increases its runtime. Three different population sizes were tested: $ps \in \{200, 500, 1000\}$
fa	Either simple Pareto-Ranking [17] ($fa = 0$) or an extended assignment process ($fa = 1$, called *variety preserving* in [52]) with sharing was applied. Sharing [30, 22] decreases the fitness of individuals which are very similar to others in the population in order to force the EA to explore many different areas in the search space.
cp	The simple convergence prevention (SCP) method proposed in [52] was either used ($cp = 0.3$) or not ($cp = 0$). SCP is a clearing approach [38, 46] applied in the objective space which discards solution candidates with equal objective values with probability cp.
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 rates do not necessarily sum up to 1, since individuals resulting from recombination may undergo mutation as well.

Table 1: The configurations used in the full-factorial experiments.

Meas.	Meaning
ar	The number of runs which found plans that completely covered all orders.
at	The *median* number of generations needed by these runs until such plans were found.
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 these runs in order 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 a 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.

Table 2: The measurements taken during the experiments.

#	mr	cr	cp	el	ps	ss	fa	ar	at	gr	gt	et	eτ	d
1.	0.8	0.4	0.3	1	1000	1	1	10	341	10	609	3078	3 078 500	15 883 km
2.	0.6	0.2	0.3	0	1000	1	1	10	502	10	770	5746	5 746 500	15 908 km
3.	0.8	0.2	0.3	1	1000	1	1	10	360	10	626	4831	4 831 000	15 929 km
4.	0.6	0.4	0.3	0	1000	1	1	10	468	10	736	5934	5 934 000	15 970 km
5.	0.6	0.2	0.3	1	1000	1	1	10	429	10	713	6236	6 236 500	15 971 km
6.	0.8	0.2	0.3	0	1000	1	1	10	375	10	674	5466	5 466 000	16 003 km
7.	0.8	0.4	0.3	1	1000	1	0	10	370	10	610	5691	5 691 500	16 008 km
8.	0.8	0.2	0.3	0	1000	0	1	10	222	10	450	6186	6 186 500	16 018 km
9.	0.8	0.4	0	0	1000	0	1	10	220	10	463	4880	4 880 000	16 060 km
10.	0.8	0.2	0	1	1000	0	0	10	277	10	506	2862	2 862 500	16 071 km
11.	0.8	0.4	0.3	0	1000	1	0	10	412	10	734	5604	5 604 000	16 085 km
12.	0.8	0.2	0.3	1	1000	0	1	10	214	10	442	4770	4 770 500	16 093 km
13.	0.8	0.2	0.3	1	1000	1	0	10	468	10	673	4970	4 970 500	16 100 km
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
181.	0.8	0.2	0	0	200	1	0	10	1286	2	6756	6773	1 354 700	20 236 km
182.	0.6	0.2	0	0	500	1	0	10	1546	1	9279	9279	4 639 500	19 529 km
183.	0.8	0.4	0.3	0	200	0	0	10	993	0	$\emptyset$	$\emptyset$	$\emptyset$	19 891 km
184.	0.8	0.4	0	0	200	0	0	10	721	0	$\emptyset$	$\emptyset$	$\emptyset$	20 352 km
185.	0.6	0.2	0	0	200	1	0	10	6094	0	$\emptyset$	$\emptyset$	$\emptyset$	23 709 km
186.	0.6	0.4	0	0	1000	1	0	0	$\emptyset$	0	$\emptyset$	$\emptyset$	$\emptyset$	∞
187.	0.8	0.4	0	0	1000	1	0	3	6191	0	$\emptyset$	$\emptyset$	$\emptyset$	∞
188.	0.8	0.4	0	0	500	1	0	4	5598	0	$\emptyset$	$\emptyset$	$\emptyset$	∞
189.	0.6	0.4	0	0	200	0	0	3	2847	0	$\emptyset$	$\emptyset$	$\emptyset$	∞
190.	0.6	0.4	0	0	200	1	0	0	$\emptyset$	0	$\emptyset$	$\emptyset$	$\emptyset$	∞
191.	0.8	0.4	0	0	200	1	0	0	$\emptyset$	0	$\emptyset$	$\emptyset$	$\emptyset$	∞
192.	0.6	0.4	0	0	500	1	0	0	$\emptyset$	0	$\emptyset$	$\emptyset$	$\emptyset$	∞

Table 3: The best and the worst evaluation results in the full-factorial tests.

The experiments indicate that a *combination* of the highest tested population size ($ps = 1000$), steady-state and elitist population treatment, SCP with rejection probability $cp = 0.3$, a sharing-based fitness assignment process, a mutation rate of 80%, and a crossover rate of 40% is able to produce the best results. We additionally applied significance tests – the sign test [47, 52] and Wilcoxon's signed rank test [47, 56, 52] – in order to check whether there also are settings of *single* parameters which generally have positive influence. On a significance level of $\alpha = 0.02$, we considered a tendency only if both (two-tailed) tests agreed. Applying the convergence prevention mechanism (SCP) [52], larger population sizes, variety preserving fitness assignment [52], elitism, and higher mutation and lower crossover rates have significantly positive influence in general.

Interestingly, the steady-state configurations lost in the significance tests against the generational ones, although the seven best-performing settings were steady-state. Here the utility of full factorial tests becomes obvious: steady-state population handling performed very well if (and only if) sharing and the SCP mechanism were applied, too. In the other cases, it led to premature convergence.

This behavior shows the following: transportation planning is a multi-modal optimization problem with a probably rugged fitness landscape [55] or with local optima which are many search steps (applications of reproduction operators) apart. Hence, applying steady-state EAs for Vehicle Routing Problems similar to the one described here can be beneficial, but only if diversity-preserving fitness assignment or selection algorithms are used in conjunction. Only then, the probability of premature convergence is kept low enough and different local optima and distant areas of the search space are explored sufficiently.

5.2 Tests with Multiple Datasets

We have run experiments with many other 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 1%, usually above 5%, and for some days even exceeding 15%. Figure 6 illustrates the best f_2-values (the total kilometers) of the individuals with the most orders satisfied in the population for two typical example evolutions.

In both diagrams, the total distance first increases as the number of orders delivered by the solution candidates rises due to the pressure from f_1. At some point, plans which are able to deliver all orders evolved and f_1 is satisfied (minimized). Now, its corresponding dimension of the objective space begins to collapse, the influence of f_2 intensifies, and the total distances of the plans decrease. Soon afterwards, the efficiency of the original plans is surpassed. Finally, the populations of the EAs converge to a Pareto frontier and no further improvements occur. In Fig. 6.1, this limit was 54 993 km, an improvement of more than 8800 km or 13.8% compared to the original distance of 63 812 km.

Each point in the graph of f_2 in the diagrams represents one point in the Pareto frontier of the corresponding generation. Fig. 6.2 illustrates one additional graph for f_2: the best plans which can be created when at most 1% of the orders are outsourced. Compared to the transportation plan including assignments for all orders which had a length of 79 464 km, these plans could reduce the distance to 74 436 km, i. e., another 7% of the overall distance could be saved. Thus, in this case, an overall reduction of around 7575 km is achieved in comparison to the original plan, which had a length of 82 013 km.

5.3 Time Consumption

One run of the algorithm (prototypically implemented in Java) for the dataset used in the full factorial tests (Section 5.1) took around three hours. For sets

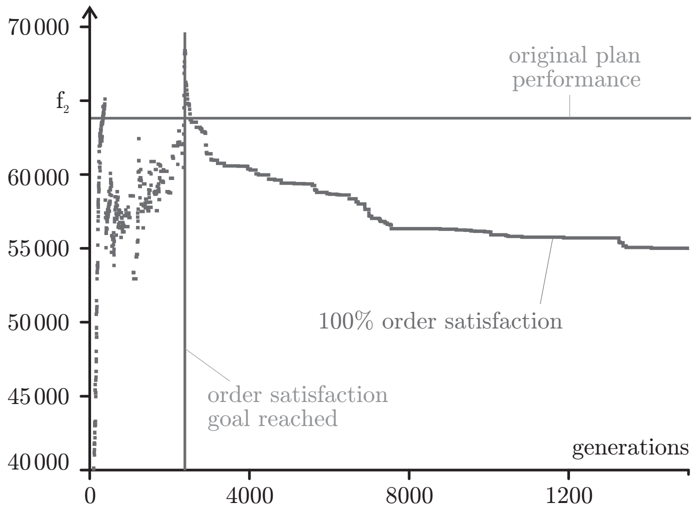

Fig. 6.1: For 642 orders (14% better).

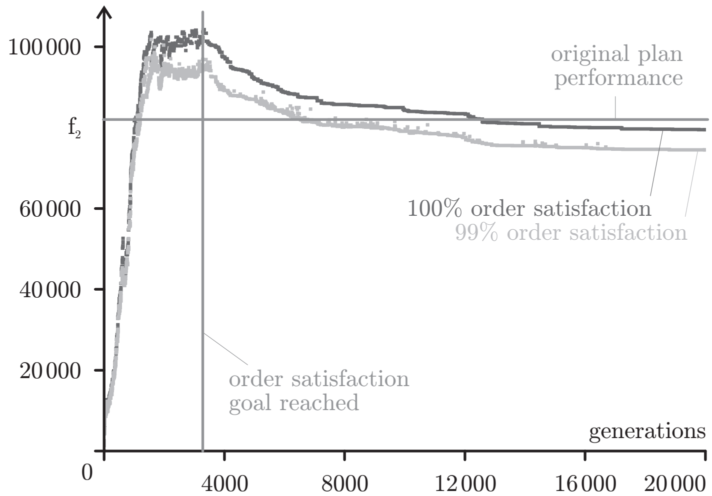

Fig. 6.2: For 1016 orders (3/10% better).

Fig. 6: Two examples for the freight plan evolution.

with around 1000 orders it still easily fulfills the requirement of delivering result within 24 hours. Runs with close to 2000 orders, however, take longer than one week (in our experiments, we used larger population sizes for them). Here, it should be pointed out that 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 parallelized and distributed over clusters [32, 53, 52]. The final implementation of the *in.west* system will incorporate distribution mechanisms and thus be able to deliver results in time for all situations.

6 Holistic Approach to Logistics

Logistics involve many aspects and the planning of efficient routes is only one of them. Customers and legislation [7], for instance, require traceability of the production data and thus, also of the goods on the road. Experts require more linkage between the traffic carriers and efficient distribution of traffic to cargo rail and water transportation.

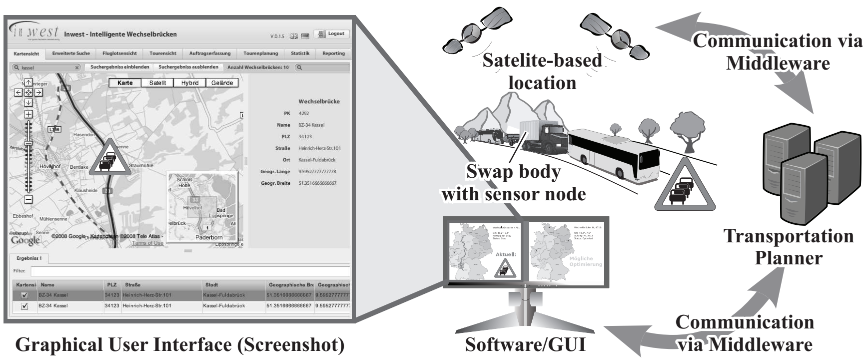

Fig. 7: An overview of the *in.west* system.

Technologies like telematics are regarded as the enablers of smart logistics which optimize the physical value chain. Only a holistic approach combining intelligent planning and such technologies can solve the challenges [49, 13] faced by logistics companies. Therefore, the focus of the *in.west* project was to combine software, telematics, and business optimization approaches into one flexible and adaptive system sketched in Figure 7.

Data from telematics become input to the optimization system which suggests transportation plans to the human operator who selects or modifies these suggestions. The final plans then, in turn, are used to configure the telematic units. A combination of both allows the customers to track their deliveries and the operator to react to unforeseen situations. In such situations, traffic jams or accidents, for instance, the optimization component can again be used to make ad-hoc suggestions for resolutions.

6.1 The Project in.west

The prototype introduced in this chapter was provided in the context of the BMWi promoted project *in.west* and will be evaluated in field test in the third quarter of 2009. The analyses take place in the area of freight transportation in the business segment *courier-*, *express-* and *parcel services* of *DHL*, the market leader in this business. Today the transport volume of this company constitutes a substantial portion of the traffic volume of the traffic carriers road, ship, and rail. Hence, a significant decrease in the freight traffic caused by *DHL* might lead to a noticeable reduction in the freight traffic volume on German roads.

The goal of this project is to achieve this decrease by utilizing information and communication technologies on swap bodies, new approaches of planning, and novel control processes. The main objective is to design a decision support tool to assist the operator with suggestions for traffic reduction and a smart swap body telematic unit.

The requirements for the *in.west* software are various, since the needs of both, the operators and the customers are to be satisfied. From their perspective, for example, a constant documentation of the transportation processes is necessary. The operators require that this documentation start with the selection, movement, and employment of the swap bodies. From the view of the customer, only tracking and tracing of the containers on the road must be supported. With the web-based user interfaces we provide, a spontaneous check of the load condition, the status of the container, and information about the carrier is furthermore possible.

6.2 Smart Information and Communication Technology

All the information required by customers and operators on the status of the freight have to be obtained by the software system first. Therefore, *in.west* also features a hardware development project with the goal of designing a telematic unit. A device called *YellowBox* (illustrated in Figure 8) was developed which enables swap bodies to transmit real time geo positioning data of the containers to a software system. The basic functions of the device are location, communication, and identification. The data received from the *Yel-*

lowBox is processed by the software for planning and controlling the swap body in the logistic network.

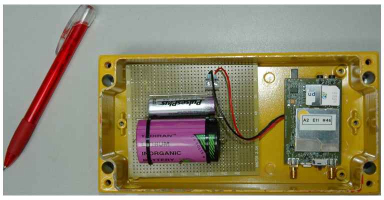

Fig. 8: The *YellowBox* – a mobile sensor node.

The *YellowBox* consist of a main board, a location unit (GPS), a communication unit (GSM/GPRS) and a processor unit. Digital input and output interfaces ensure the scalability of the device, e.g., for load volume monitoring. Swap bodies do not offer reliable power sources for technical equipment like telematic systems. Thus, the *YellowBox* has been designed as a sensor node which uses battery current [40].

One of the most crucial criteria for the application of these sensor nodes in practice was long battery duration and thus, low power consumption. The *YellowBox* is therefore turned on and off for certain time intervals. The software system automatically configures the device with the route along which it will be transported (and which has been evolved by the planner). In predefined time intervals, it can thus check whether or not it is "on the right track". Only if location deviations above a certain threshold are detected, it will notify the middleware. Also, if more than one swap body is to be transported by the same vehicle, only one of them needs to perform this notification. With these approaches, communication – one of the functions with the highest energy consumption – is effectively minimized.

7 Conclusions

In this chapter, we presented the *in.west* freight planning component which utilizes an Evolutionary Algorithm with intelligent reproduction operations for general transportation planning problems. The approach was tested rigorously on real-world data from the *in.west* partners and achieved excellent results. It has been integrated as a constituting part of the holistic *in.west* logistics software system.

We presented this work at the EvoTRANSLOG'09 workshop in Tübingen [54]. One point of the very fruitful discussion there was the question why we did not utilize heuristics to create some initial solutions for the EA. We intentionally left this for our future work for two reasons: First, we fear that creating such initial solutions may lead to a decrease of diversity in the population. In Section 5.1 we showed that diversity is a key to finding good solutions to this class of problem. Second, as can be seen in the diagrams provided in Figure 6, finding initial solutions where all orders are assigned to routes is not the time consuming part of the EA – optimizing them to plans with a low total distance is. Hence, incorporating measures for distribution and efficient parallelization may be a more promising addition to our approach. If a cluster of, for instance, twenty computers is available, we can assume that distribution according to client-server or island model schemes [34, 29, 53, 52] will allow us to decrease the runtime to at least one tenth of the current value. Nevertheless, testing the utility of heuristics for creating the initial population is on our agenda, too.

In the current phase, some of the functions of the component still work on a rather prototypical level. They will be updated in order to make the system ready for the field test in Fall 2009. We will therefore improve the support for parallelization and integrate components for distributing the computational load. As already pointed out, this is likely the best way to resolve the remaining timing issues for very large datasets. Additionally, features like online updates of the distance matrix which is used to both, to compute f_2 and also to determe 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. It should be noted that even with such a system, the human operator cannot be replaced. There are always constraints and pieces of information which cannot be employed in even the most advanced automated optimization process. Hence, the solutions generated by our transportation planner are *suggestions* rather than doctrines. They are displayed to the operator and she may modify them according to her needs.

Looking forward to deploying this new system in the computer centers of the project partners, we are confident that *in.west* will fulfill their expectations.

Acknowledgements
The research work presented here has been sponsored by the German Federal Ministry of Economics and Technology. Also, we wish to thank Raymond Chiong for his helpful comments and careful proofreading.

References

1. 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, *Lecture Notes in Computer Science (LNCS)*, vol. 3004/2004, pp. 11–20. Springer (2004). DOI 10.1007/b96499
2. Alba, E., Dorronsoro, B.: Computing nine new best-so-far solutions for capacitated vrp with a cellular genetic algorithm. Information Processing Letters **98**, 225–230 (2006). DOI 10.1016/j.ipl.2006.02.006
3. 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), 360–376 (2000). DOI 10.1016/S0377-2217(99)00170-8
4. Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D., Rinaldi, G.: Computational results with a branch and cut code for the capacitated vehicle routing problem. Research Report 949-M, Universite Joseph Fourier, Grenoble, France (1995)
5. Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press (1996)
6. 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), 109–122 (1997)
7. van Betteray, K.: Gesetzliche und handelsspezifische anforderungen an die rückverfolgung. In: Vorträge des 7. VDEB-Infotags 2004. VDEB Verband IT-Mittelstand e.V. (2004). EU Verordnung 178/2002
8. Box, G.E.P., Hunter, J.S., Hunter, W.G.: Statistics for Experimenters: Design, Innovation, and Discovery. John Wiley & Sons (2005)
9. Bräysy, O.: Genetic algorithms for the vehicle routing problem with time windows. Arpakannus – Newsletter of the Finnish Artificial Intelligence Society (FAIS) **1**, 33–38 (2001). Special issue on Bioinformatics and Genetic Algorithms.
10. 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), 211–237 (2002). DOI 10.1007/BF02579017
11. 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. Ph.D. thesis, University of Antwerp, RUCA, Belgium (1994)
12. Bullnheimer, B., Hartl, R.F., Strauss, C.: An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research **89**, 319–328 (1999). DOI 10.1023/A:1018940026670
13. Bundesministerium für Verkehr, Bau- und Stadtentwicklung: Verkehr in Zahlen 2006/2007. Deutscher Verkehrs-Verlag GmbH, Hamburg, Germany (2006)
14. Bundesministerium für Wirtschaft und Technologie: Mobilität und Verkehrstechnologien das 3. Verkehrsforschungsprogramm der Bundesregierung. BMWi, Öffentlichkeitsarbeit, Berlin, Germany (2008)
15. CEN/TC 119: Swap bodies – non-stackable swap bodies of class C – dimensions and general requirements. EN 284, CEN-CEN ELEC, Brussels, Belgium (2006)
16. Ceollo Coello, C.A.: A short tutorial on evolutionary multiobjective optimization. In: E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, D. Corne (eds.) Evolutionary Multi-Criterion Optimization, Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01), *Lecture Notes in Computer Science (LNCS)*, vol. 1993/2001, pp. 21–40. Springer-Verlag, Berlin (2001)
17. Ceollo Coello, C.A., Lamont, G.B., van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, *Genetic and Evolutionary Computation*, vol. 5, second edn. Kluwer Academic Publishers / Springer (1st ed: 2002, 2nd ed: 2007). DOI 10.1007/978-0-387-36797-2

18. Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In: N. Christofides, A. Mingozzi, P. Toth, C. Sandi (eds.) Combinatorial Optimization, chap. 11, pp. 315–338. John Wiley & Sons Ltd (1979)
19. Confessore, G., Galiano, G., Stecca, G.: An evolutionary algorithm for vehicle routing problem with real life constraints. In: M. Mitsuishi, K. Ueda, F. Kimura (eds.) Manufacturing Systems and Technologies for the New Frontier – The 41st CIRP Conference on Manufacturing Systems, pp. 225–228. Springer (2008). DOI 10.1007/978-1-84800-267-8_46
20. 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), pp. 376–383. IEEE Computer Society (2002). DOI 10.1109/EMPDP.2002.994313
21. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Management Science **6**(1), 80–91 (1959). DOI 10.1287/mnsc.6.1.80
22. Deb, K., Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In: J.D. Schaffer (ed.) Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 42–50. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)
23. Díaz, B.D.: Known best results (2007). Online available at `http://neo.lcc.uma.es/radi-aeb/WebVRP/results/BestResults.htm` [accessed 2007-12-28]
24. 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, pp. 11–20. Springer-Verlag (2002). DOI 10.1007/3-540-46004-7_2
25. Domschke, W.: Logistik, Rundreisen und Touren, fourth edn. Oldenbourgs Lehr- und Handbücher der Wirtschafts- u. Sozialwissenschaften. Oldenbourg Verlag (1997)
26. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & Operations Research **13**(5), 533–549 (1986). DOI 10.1016/0305-0548(86)90048-1
27. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA (1989)
28. Golden, B.L., Wasil, E.A., James P. Kelly and, I.M.C.: The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results. In: T.G. Crainic, G. Laporte (eds.) Teodor Gabriel Crainic and Gilbert Laporte, Center for Research on Transportation 25th Anniversary Series, 1971–1996, chap. 2, pp. 33–56. Kluwer / Springer, Boston, USA (1998)
29. Gorges-Schleuter, M.: Explicit parallelism of genetic algorithms through population structures. In: H.P. Schwefel, R. Männer (eds.) PPSN I: Proceedings of the 1st Workshop on Parallel Problem Solving from Nature, *Lecture Notes in Computer Science (LNCS)*, vol. 496/1991, pp. 150–159. Springer (1990)
30. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The University of Michigan Press, Ann Arbor (1975). Reprinted by MIT Press, April 1992, NetLibrary, Inc
31. Jih, W., Hsu, J.: Dynamic vehicle routing using hybrid genetic algorithms. In: IEEE International Conference on Robotics and Automation, pp. 453–458 (1999). DOI 10.1109/ROBOT.1999.770019
32. Luke, S., Panait, L., Balan, G., Paus, S., Skolicki, Z., Bassett, J., Hubley, R., Chircop, A.: Ecj: A java-based evolutionary computation research system (2006). Version 18. For more information see `http://cs.gmu.edu/~eclab/projects/ecj/` [accessed 2007-07-10]
33. Machado, P., Tavares, J., Pereira, F.B., Costa, E.J.F.: Vehicle routing problem: Doing it the evolutionary way. In: W.B. Langdon, E. Cantú-Paz, K.E. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull,

M.A. Potter, A.C. Schultz, J.F. Miller, E.K. Burke, N. Jonoska (eds.) GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, p. 690. Morgan Kaufmann Publishers Inc. (2002)

34. Martin, W.N., Lienig, J., Cohoon, J.P.: Island (migration) models: Evolutionary algorithms based on punctuated equilibria. In: T. Bäck, D.B. Fogel, Z. Michalewicz (eds.) Handbook of Evolutionary Computation, Computational Intelligence Library, chap. 6.3. Oxford University Press (1997)
35. Ombuki-Berman, B.M., Hanshar, F.: Using genetic algorithms for multi-depot vehicle routing. In: Bio-inspired Algorithms for the Vehicle Routing Problem, pp. 77–99. Springer (2009). DOI 10.1007/978-3-540-85152-3_4
36. Pankratz, G., Krypczyk, V.: Benchmark data sets for dynamic vehicle routing problems (2007). Data sets online available at `http://www.fernuni-hagen.de/WINF/inhfrm/benchmark_data.htm` [accessed 2008-10-27]
37. Pereira, F.B., Tavares, J. (eds.): Bio-inspired Algorithms for the Vehicle Routing Problem, *Studies in Computational Intelligence*, vol. 161. Springer Berlin / Heidelberg (2009). DOI 10.1007/978-3-540-85152-3
38. Pétrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation, CEC96, pp. 798–803. IEEE Press, Piscataway, NJ (1996). DOI 10.1109/ICEC.1996.542703
39. 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 (2009)
40. Podlich, A., Weise, T., Menze, M., Gorldt, C.: Intelligente wechselbrückensteuerung für die logistik von morgen. In: M. Wagner, D. Hogrefe, K. Geihs, K. David (eds.) First Workshop on Global Sensor Networks (GSN'09) (2009). Electronic Communications of the EASST (ECASST), ISSN 1863-2122, volume 17, part Global Sensor Networks (GSN 2009), The European Association of Software Science and Technology
41. Potvin, J.Y.: A review of bio-inspired algorithms for vehicle routing. In: Bio-inspired Algorithms for the Vehicle Routing Problem, pp. 1–34. Springer (2009). DOI 10.1007/978-3-540-85152-3_1
42. Radcliffe, N.J.: The algebra of genetic algorithms. Annals of Mathematics and Artificial Intelligence **10**(4), 339–384 (1994). DOI 10.1007/BF01531276
43. Ralphs, T.: Vehicle routing data sets (2003). Online available at `http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/` [accessed 2009-04-08]
44. Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter, L.E.: On the capacitated vehicle routing problem. Mathematical Programming **94**(2–3), 343–359 (2003). DOI 10.1007/s10107-002-0323-0
45. von Randow, M.: Güterverkehr und logistik als tragende säule der wirtschaft zukunftssicher gestalten. In: H. Baumgarten (ed.) Das Beste Der Logistik: Innovationen, Strategien, Umsetzungen, pp. 49–53. Bundesvereinigung Logistik (BVL), Springer, Berlin/Heidelberg, Germany (2008)
46. Sareni, B., Krähenbühl, L.: Fitness sharing and niching methods revisited. IEEE Transactions on Evolutionary Computation **2**(3), 97–106 (1998). DOI 10.1109/4235.735432
47. Siegel, S., Castellan Jr., N.J.: Nonparametric Statistics for The Behavioral Sciences. Humanities/Social Sciences/Languages. McGraw-Hill (1988)
48. Sigurjónsson, K.: Taboo search based metaheuristic for solving multiple depot vrppd with intermediary depots. Master's thesis, Informatics and Mathematical Modelling, IMM, Technical University of Denmark, DTU (2008). Online available at `http://orbit.dtu.dk/getResource?recordId=224453&objectId=1&versionId=1` [accessed 2009-04-09]
49. Steierwald, G., Künne, H.D., Vogt, W.: Stadtverkehrsplanung: Grundlagen, Methoden, Ziele, 2., neu bearbeitete und erweiterte auflage edn. Springer, Berlin, Germany (2005). DOI 10.1007/b138349

50. Taillard, É.D.: Parallel iterative search methods for vehicle routing problems. Networks **23**(8), 661–673 (1993). DOI 10.1002/net.3230230804
51. Thangiah, S.R.: Vehicle routing with time windows using genetic algorithms. In: Practical Handbook of Genetic Algorithms: New Frontiers, pp. 253–277. CRC (1995)
52. Weise, T.: Global Optimization Algorithms – Theory and Application, second edn. (2010). Online available at `http://www.it-weise.de/` [accessed 2010-06-08]
53. Weise, T., Geihs, K.: DGPF – An Adaptable Framework for Distributed Multi-Objective Search Algorithms Applied to the Genetic Programming of Sensor Networks. In: B. Filipič, J. Šilc (eds.) Proceedings of the Second International Conference on Bioinspired Optimization Methods and their Applications (BIOMA 2006), pp. 157–166. Jožef Stefan Institute (2006)
54. Weise, T., Podlich, A., Reinhard, K., Gorldt, C., Geihs, K.: Evolutionary freight transportation planning. In: EvoTRANSLOG, 3rd European Workshop on Evolutionary Computation in Transportation and Logistics, Applications of Evolutionary Computing – Proceedings of EvoWorkshops 2009, *Lecture Notes in Computer Science (LNCS)*, vol. 5484/2009, pp. 768–777. Springer (2009). DOI 10.1007/978-3-642-01129-0_87
55. Weise, T., Zapf, M., Chiong, R., Nebro, A.J.: Why is optimization difficult? In: R. Chiong (ed.) Nature-Inspired Algorithms for Optimisation, *Studies in Computational Intelligence*, vol. 193, chap. 1, pp. 1–50. Springer (2009). DOI 10.1007/978-3-642-00267-0_1
56. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**(6), 80–83 (1945)
57. Yates, F.: The Design and Analysis of Factorial Experiments. Imperial Bureau of Soil Science, Commonwealth Agricultural Bureaux (1937). Tech. Comm. No. 35
58. 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, pp. 176–183. IEEE Computer Society (2003). DOI 10.1109/TAI.2003.1250187

```
@incollection{WPG2009SRWVRPWEA,
  title     = {Solving Real-World Vehicle Routing Problems with
               Evolutionary Algorithms},
  author    = {Thomas Weise and Alexander Podlich and Christian Gorldt},
  booktitle = {Natural Intelligence for Scheduling, Planning and
               Packing Problems},
  editor    = {Raymond Chiong and Sandeep Dhakal},
  isbn      = {978-3-642-04038-2},
  series    = {Studies in Computational Intelligence},
  ISSN      = {1860-949X (Print) 1860-9503 (Online)},
  volume    = {250/2009},
  publisher = {Springer Berlin / Heidelberg},
  year      = {2009},
  pages     = {29--53},
  doi       = {10.1007/978-3-642-04039-9_2},
  keywords  = {Evolutionary Algorithm, Intelligent Search Operators,
               Phenotype, Transportation Planning, Vehicle Routing Problem,
               VRP, MDVRP, DVRP, CVRP, VRPTW, VRPPD, YellowBox, GPS,
               Mobile Phone, GPRS, DHL}
  url       = {htpp://www.it-weise.de/documents/files/WPG2009SRWVRPWEA.pdf},
}
```